

# Indirect Supervision for the Determination and Structural Analysis of Nominal Compounds

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik  
der Universität Stuttgart zur Erlangung der Würde eines Doktors  
der Philosophie (Dr. phil.) genehmigte Abhandlung.

Vorgelegt von  
Patrick René Ziering  
aus Göppingen

Hauptberichter	Dr. Lonneke van der Plas
Mitberichter	PD Dr. Sabine Schulte im Walde
Mitberichter	Prof. Dr. Sebastian Padó

Tag der mündlichen Prüfung: 21. Dezember 2017

Institut für Maschinelle Sprachverarbeitung  
der Universität Stuttgart

2018



## Erklärung (Statement of Authorship)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet.

Die eingereichte Arbeit ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen. Sie ist weder vollständig noch in Teilen bereits veröffentlicht.

Die beigefügte elektronische Version stimmt mit dem Druckexemplar überein.

I hereby declare that this text is the result of my own work and that I have not used sources without declaration in the text. Any thoughts from others or literal quotations are clearly marked.

The thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

The enclosed electronic version is identical to the printed versions.

---

Datum, Ort

---

Unterschrift



# Abstract

Determining and analyzing **lexemes**, the building blocks of natural language, is the starting point for many **Natural Language Processing (NLP)** tasks. In this thesis, we try to tackle the challenge of analyzing **complex lexemes**, composed of several **atomic lexemes**. The major representative of **complex lexemes**, the **compound**, is an important subject of study in theoretical linguistics, as it is located at the boundary between syntax (e.g., the phrase *French car*) and lexicon (e.g., the **lexeme** *French toast*). **Compounds** are abundant in many languages and occur in various embodiments (e.g., *flowerpot*, *flower-pot* or *flower pot*). **Compounding** is one of the most productive **word** formation types and ‘almost any pair of English nouns can be combined to produce a valid, if not always sensible, **compound**’ (Ó Séaghdha, 2008). German corpus studies revealed that almost half (47%) of the **word types** were **compounds**, whereas most **compounds** are infrequent (83% of the **compounds** had a corpus frequency of 5 or lower) (Baroni et al., 2002). Being abundant as a phenomenon but scarce in terms of individual examples (i.e., the combination of high **type** frequency and low **token** frequency) makes the analysis of **compounds** particularly problematic for statistical techniques that need high **token** frequencies to make accurate predictions. As a consequence, data sparsity is expected to lead to low performance.

There is a vast amount of previous work in **NLP** addressing compositional **compound analysis** (including the **identification**, **structural analysis** and semantic analysis of **compounds**), which is recursively based on the analysis of their **immediate constituents**. Most previous methods for automatic **compound analysis** use manual resources such as morphological analyzers (e.g., Fritzing and Fraser (2010)) or hand-crafted transformation rules (e.g., Stymne (2008)), or are based on supervised approaches relying on manual training data (e.g., Vadas and Curran (2007b)). Thus, most previous work cannot be applied easily to new domains and languages.

The correct **analysis of compounds** is important for many **NLP** tasks, such as **Machine Translation (MT)** (Johnston and Busa, 1999, Navigli et al., 2003). The accurate translation of **compounds** is non-trivial, because we find a large amount of variation in the way languages deal with **compounding**. Some languages such as German use **closed compounding** (i.e., they create **one-word compounds** e.g., *Todesstrafe* ‘death penalty’), whereas others do not. In Romance languages, such as French, **compounds** are not as productive, instead **complex nominals** (e.g., *peine de mort* ‘death penalty’) are used.

In this thesis, we address the analysis of **nominal compounds** in terms of **compoundhood** determination and **structural analysis**. Despite their abundance, the definition (and even the existence) of **compounds** is controversially discussed in linguistics literature (Lieber and Štekauer, 2009). We inspect the relevance of various established **linguistic criteria** for **compoundhood** and provide new insights on the phenomenon.

In our approaches for analyzing compounds structurally, we aim to avoid relying on direct supervision in terms of hand-labeled training data or manual resources, and instead focus on **indirect supervision** by means of *naturally occurring supervision* (Snyder and Barzilay, 2010), the main reason being to produce language-independent, resource-lean applications. A **cross-lingual** corpus study on English **nominal compounds** revealed the large space of surface variations across languages, which allows for **cross-lingual supervision** for **compound analysis**. We present two analysis tasks that enjoy **cross-lingual supervision**. Firstly, we exploit **cross-lingual** evidence for the task of **compound identification**. For example, knowing whether *French teacher* is translated to German as *Französischlehrer* or as *französischer Lehrer* is beneficial for determining the **compoundhood status** and the meaning (i.e., ‘a person teaching French’ vs. ‘a teacher having a French nationality’). Secondly, **cross-lingual** support is used for the task of **compound parsing**. For example, for the English **three-Noun Compound (3NC)** *human<sub>A</sub> rights<sub>B</sub> violation<sub>C</sub>* being translated to German as *Verletzung<sub>C</sub> der Menschenrechte<sub>A B</sub>*, the fact that the **constituent equivalent** of *violation<sub>C</sub>*, *Verletzung<sub>C</sub>*, is separated from the other **constituent equivalents**, points us in the direction of a LEFT-branched structure, i.e., [*human<sub>A</sub> rights<sub>B</sub>*] *violation<sub>C</sub>*.

Moreover, we address the task of **compound splitting**. Here, we exploit a form of **indirect supervision** that relies on monolingual morphological regularities between regular **word inflection** and **compound-specific constituent inflection** (e.g., **linking elements**), thereby eschewing a limitation of **cross-lingual supervision**, i.e., the dependence on **parallel data**. For example, the plural form of a **lexeme** often conforms with its **constituent form**, e.g., the German *Hühner* ‘chicken<sub>plural</sub>’ as in *Hühner | suppe* ‘chicken soup’.

As a final result, we observe that our proposed methods achieve competitive performance that is state-of-the-art within the scope of **indirectly supervised** methods. Moreover, the nature of the approaches, which are for the most part motivated by linguistic theory, shed light on the complex phenomenon of **compoundhood** in **cross-lingual** as well as monolingual settings.

# Deutsche Zusammenfassung

Das Bestimmen sowie die Analyse von Lexemen, die Bestandteile natürlicher Sprache, bilden die Ausgangslage vieler Methoden der Maschinellen Sprachverarbeitung (MSV). In der vorliegenden Dissertation stellen wir uns der Herausforderung, komplexe Lexeme, die aus atomaren Lexemen aufgebaut sind, zu analysieren. Die wichtigste Klasse von komplexen Lexemen, das Kompositum, ist ein grundlegendes Thema in der theoretischen Linguistik, da es häufig als eine Mischform aus syntaktischen (z.B. die Phrase *French car* ‘französisches Auto’) und lexikalischen Aspekten (z.B. das Nominalkompositum *French toast*) angesehen wird. Komposita sind ein sehr häufiges Phänomen in vielen Sprachen und treten in zahlreichen Ausprägungen auf, etwa *flowerpot*, *flower-pot* oder *flower pot*. Die Kompositabildung ist eine der produktivsten Wortbildungstypen; es kann nahezu jedes Paar aus englischen Nomen zu einem gültigen, wenn auch nicht immer sinnhaften, Kompositum kombiniert werden (Ó Séaghdha, 2008). Deutsche Korpusstudien zeigten dass fast die Hälfte (47%) aller Wort-Typen Komposita sind, wohingegen die meisten Komposita selten auftreten (83% der Komposita haben eine Korpusfrequenz von 5 oder weniger) (Baroni et al., 2002). Häufig als Phänomen aber selten als individuelles Lexem (d.h., die Kombination aus hoher Typ-Frequenz und niedriger Token-Frequenz) bedeutet besonders für statistische Techniken, die für eine präzise Vorhersage eine hohe Token-Frequenz benötigen, eine besondere Problematik. Dies hat zur Folge, dass Datenknappheit häufig zu schwächeren Ergebnissen führt.

Es gibt eine große Anzahl an bisheriger Arbeiten in MSV, die eine kompositionelle Komposita-Analyse (inkl. dem Bestimmen und der strukturellen sowie semantischen Analyse von Komposita) behandeln, d.h., die Analyse basiert rekursiv auf den Analysen der unmittelbaren Konstituenten. Die meisten bisherigen Methoden der automatischen Komposita-Analyse verwenden manuelle Ressourcen wie etwa morphologische Analysesysteme (z.B. Fritzingler and Fraser (2010)) oder handgeschriebene Transformationsregeln (z.B. Stymne (2008)), oder basieren auf überwachten Ansätzen, die auf manuelle Trainingsdaten setzen (z.B. Vadas and Curran (2007b)). Daher können die meisten bisherigen Arbeiten nicht ohne Weiteres auf neue Domänen und Sprachen angewendet werden.

Eine korrekte Komposita-Analyse ist wichtig für viele MSV Methoden, wie etwa Maschinelle Übersetzung (Johnston and Busa, 1999, Navigli et al., 2003). Die präzise Übersetzung von Komposita ist nicht trivial, da es eine große Vielfalt darin gibt, wie unterschiedliche Sprachen mit Kompositabildung umgehen. Manche Sprachen, etwa Deutsch, verwenden *geschlossene Komposita* (also Ein-Wort-Komposita wie *Todesstrafe*), wohingegen andere Sprachen diese nicht verwenden. In romanischen Sprachen, etwa Französisch, ist die Kompositabildung weniger produktiv. Stattdessen werden komplexe Nominale (z.B. *peine de mort* ‘Todesstrafe’) verwendet.

In der vorliegenden Dissertation behandeln wir die Analyse von Nominalkomposita in Bezug auf die Bestimmung der Kompositionshaftigkeit und der Struktur. Trotz ihrer Häufigkeit wird sowohl die Definition als auch selbst die Existenz von Komposita kontrovers in der linguistischen Literatur diskutiert (Lieber and Štekauer, 2009). Wir untersuchen die Wichtigkeit von zahlreichen bekannten linguistischen Kriterien zur Kompositionshaftigkeit und finden neue Erkenntnisse über das Phänomen heraus.

Unsere Ansätze zur strukturellen Analyse von Komposita vermeiden die Abhängigkeit von direkter Überwachung bzgl. hand-annotierter Trainingsdaten oder manueller Ressourcen. Stattdessen verwenden wir die indirekte Überwachung als ein Mittel der *natürlich vorkommenden Überwachung* (Snyder and Barzilay, 2010), mit dem Ziel, Anwendungen zu entwickeln, die unabhängig von Sprache und manuellen Ressourcen arbeiten.

Eine sprachübergreifende Korpusstudie auf englischen Nominalkomposita veranschaulichte die vielzähligen Möglichkeiten, wie Nominalkomposita in anderen Sprachen realisiert werden können. Diese Form der Variation ermöglicht die Komposita-Analyse durch sprachübergreifende Überwachung. Wir behandeln zwei Analyse-Aufgaben, die sprachübergreifend überwacht werden. Zum einen verwenden wir sprachübergreifende Information für die Komposita-Bestimmung. Zum Beispiel kann die deutsche Übersetzung von *French teacher* Aufschlüsse auf den Status der Kompositionshaftigkeit sowie auf die Bedeutung geben, etwa *Französischlehrer* als Kompositum oder *französischer Lehrer* als Nominalphrase. Zum anderen kann man das Parsing von Komposita durch sprachübergreifende Ansätze unterstützen. Zum Beispiel das englische Kompositum *human<sub>A</sub> rights<sub>B</sub> violation<sub>C</sub>* kann mit der deutschen Übersetzung *Verletzung<sub>C</sub> der Menschenrechte<sub>A B</sub>* strukturiert werden. Die Tatsache, dass das Äquivalent der dritten Konstituenten, *Verletzung<sub>C</sub>*, vom Rest durch ein Funktionswort getrennt ist, deutet auf eine links-verzweigte Struktur, [*human<sub>A</sub> rights<sub>B</sub>]* *violation<sub>C</sub>*, hin.

Des Weiteren behandeln wir die Kompositazerlegung. Hierbei basieren unsere Ansätze auf einer Form der indirekten Überwachung, die auf monolingualen morphologischen Analogien beruht, und zwar zwischen regulärer Flexion und der Kompositionsspezifischen Konstituenten-Flexion (z.B., Fugen-Elemente). Diese Form der indirekten Überwachung vermeidet die Einschränkung sprachübergreifend überwachter Systeme, nämlich die Abhängigkeit von parallelen Daten. So entspricht beispielsweise die Pluralform eines Lexems häufig ihrer Konstituentenform, etwa das deutsche pluralisierte Lexem *Hühner* im Kompositum *Hühner | suppe*.

Zum Ende unserer Forschung kommen wir zu dem Ergebnis, dass unsere vorgeschlagenen Ansätze eine konkurrenzfähig Leistung erzielen, die dem heutigen Stand der Technik für indirekt überwachte Systeme entspricht. Zudem bringen die linguistisch motivierten Ansätze neue Erkenntnisse über das komplexe Phänomen von Komposita sowohl im sprachübergreifenden wie auch im monolingualen Kontext.



# Acknowledgements

The past seven years as PhD student have been a time full of highs and lows. There were hard periods with accumulations of failing experiments but there were also periods of successful streaks without any submission rejection. Both the good and bad times shaped my scientific mindset. It was a time of learning, experimenting and writing. I have learned to deal with periods of stress (e.g., when working towards a submission deadline) and to keep motivated after having read a pessimistic submission review.

## People and Groups

There is a big number of people to whom I am deeply indebted and who have contributed to getting me finished this thesis.

My honest and deepest thanks go to **Lonneke van der Plas**. I never expected to find such a perfect supervisor in her. As a junior professor at the IMS, she agreed to take me over from Hinrich Schütze in 2013. We had many highly interesting discussions about linguistics, such as phenomena in compounding, and about writing a convincing conference paper and PhD thesis. While I was being lost in experimental details, Lonneke pointed at the relevant research questions and never lost the overview of our goals. I deeply thank Lonneke for keeping me as PhD student after she has moved to Malta. In spite of the great geographical distance, we had an excellent communication by email or video chat. I am most grateful to Lonneke for the huge amount of time and effort she spent in our meetings and for preparing publications. This was overwhelming. I highly appreciate all work she has done for me.

I am very grateful to **Hinrich Schütze** for being my first supervisor at the IMS. He valued my great success as diploma student and allowed me to start researching in a highly qualified faculty of statistical NLP. Hinrich spent a lot of time in reading my draft papers and in providing important comments. I appreciate his patience in explaining me all basics about research. Hinrich organized many administrative issues for me.

Another thank you goes to **Sabine Schulte im Walde** for always being open for questions. I highly appreciate the time and effort Sabine spent in reading draft papers and giving me useful comments. Sabine also helped me structure my PhD thesis. Finally, she supported me in organizing my PhD defense. Sabine integrated me into the **SemRel group**. Being part of this group allowed me to present results for experiments and ideas for new research directions. I received very helpful feedback and inspiration from all group members.

I am also grateful to my **PhD examiners**, **Lonneke van der Plas**, **Sabine Schulte im Walde** and **Sebastian Padó** for reading the large thesis and providing comments.

I also thank **Jonas Kuhn** for providing a nice atmosphere at the IMS that allowed me to research and write my thesis up in a pleasant environment.

## Acknowledgements

Many thanks to all **secretaries** that helped me with administrative issues, in particular **Barbara Schäfer** (for managing my vacation and sick days even after the TOPAS project) and **Sabine Mohr** (for looking after my SFB employment contracts, the HiWi contracts and for helping me with other issues such as the business trip applications).

A great thank you goes to my colleague and good friend **Andrea Glaser**. She already accompanied me during my courses and exams as a diploma student. Andrea helped me through hard times with long conversations and motivated me to continue my PhD studies and to never give up.

Another great thank you goes to my office mate **Stefan Müller**. I enjoyed the time when we shared the office. I am grateful for so many interesting and entertaining discussions, some of which have also led to new research questions (e.g., how to deal with constituent inflection in compound splitting). I wish him all the best for recovering from his serious disease. Keep fighting all the time!

A special thank you goes to my **Mensa group** including **Max Kisselew**, **Ina Rösiger**, **Andrea Glaser**, **Stefan Müller**, **Yvonne Viesel** and others. I am very thankful for integrating me into the Mensa group and for allowing me to participate in so many interesting discussions about linguistics, politics and everyday questions.

As for other colleagues at IMS and other departments, I thank **Glorianna Jagfeld** for her great job with our ACL paper about the usage of RTE as extrinsic evaluation task for compound splitting (Jagfeld et al., 2017). I also thank **Charles Jochim** for solving so many problems I have run into when programming in Java and Python, and **Marion Di Marco** and **Fabienne Cap** for the interesting discussions about compound splitting and for providing me with the most recent versions of their splitting methods. I thank **Britta Zeller** for inspiring me to work on RTE for compound splitting, and for supporting us in running the RTE algorithms on the Excitement Open Platform (EOP).

Finally, I thank the **anonymous reviewers** of all submitted conference papers for their honest, constructive and helpful feedback.

### Projects and Funds

Supported and funded by different research projects, I was able to get in touch with various fields of NLP research.

I am grateful to the project **WordGraph** funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) and to the project **TOPAS** (Tool platform for intelligent Patent Analysis and Summarization) and the European Commission with its FP7-SME Program. I thank all colleagues for their support. The work in TOPAS provided many experiences in developing and presenting an industrial product in an international environment.

I am also grateful to the DFG as part of the Collaborative Research Centre (Sonderforschungsbereich, SFB) 732. Special thanks go to the **SFB 732 project D11** entitled “*A Crosslingual Approach to the Analysis of Compound Nouns*” that guided my research on compounding.

### Dedication

This thesis is dedicated to my **parents** for giving me the chance of doing my PhD, for supporting me every day during this long time period and for never losing faith in me.

# Table of Contents

Title page	i
Erklärung (Statement of Authorship)	iii
Abstract	v
Deutsche Zusammenfassung	vii
Acknowledgements	ix
Table of Contents	xi
<b>Part A Preface</b>	<b>1</b>
<b>1 Introduction to the Thesis</b>	<b>3</b>
1.1 Thematic Structure of the Thesis . . . . .	5
1.2 Motivation of the Thesis . . . . .	7
1.2.1 Motivation for Analyzing Compounds . . . . .	7
1.2.2 Motivation for our Methodology . . . . .	12
1.3 Overview of the Main Research Questions . . . . .	16
1.3.1 Compoundhood . . . . .	16
1.3.2 Indirect Supervision and Avoidance of Manual resources . . . . .	17
1.4 Main Contributions of the Thesis . . . . .	18
1.4.1 New Insights about the Notion of Compoundhood . . . . .	18
1.4.2 Potential of Cross-lingual Evidence on Association Strength for Compound Analysis . . . . .	18
1.4.3 Utilization of Language-independent Morphological Regularities . . . . .	19
1.4.4 Lexical Resources . . . . .	20
1.5 Outline of the Thesis . . . . .	21
<b>Part B Nature of Compounds</b>	<b>25</b>
<b>2 Introduction to Nature of Compounds</b>	<b>27</b>
2.1 Basic Description . . . . .	27

Table of Contents

2.2	Outline . . . . .	28
<b>3</b>	<b>General Aspects</b>	<b>31</b>
3.1	Multi-Word Expressions . . . . .	31
3.2	Naming Convention . . . . .	32
	3.2.1 Inconsistent Naming in Previous Literature . . . . .	32
	3.2.2 Naming in this Thesis . . . . .	32
3.3	Productivity and Corpus Distribution . . . . .	33
3.4	Functions of Compounds . . . . .	34
3.5	Spelling . . . . .	35
	3.5.1 Closed Compounds . . . . .	35
	3.5.2 Hyphenated Compounds . . . . .	35
	3.5.3 Open Compounds . . . . .	36
	3.5.4 Mixed Spelling Forms . . . . .	36
3.6	Constituents . . . . .	36
	3.6.1 Head . . . . .	37
	3.6.2 Headedness . . . . .	37
	3.6.3 Modifier . . . . .	38
	3.6.4 Complex Compounds and their Structure . . . . .	38
3.7	Compound Classes . . . . .	40
	3.7.1 Universal Taxonomy . . . . .	40
	3.7.2 Neoclassical Compounds . . . . .	42
	3.7.3 Phrasal Compounds . . . . .	43
	3.7.4 Other Classes of Compounds . . . . .	44
	3.7.5 Compound Classes in this Thesis . . . . .	45
3.8	Compound Semantics . . . . .	45
	3.8.1 Compositionality . . . . .	45
	3.8.2 Semantic Relation . . . . .	46
	3.8.3 Semantic Indeterminacy . . . . .	46
3.9	Compounding across Languages . . . . .	50
	3.9.1 English . . . . .	50
	3.9.2 German . . . . .	51
	3.9.3 Dutch . . . . .	54
	3.9.4 Afrikaans . . . . .	57
<b>4</b>	<b>The Controversy of the Definition of Compounds</b>	<b>61</b>
4.1	Various Ways of Compound Definition . . . . .	61
4.2	The Key Issues for the Compound Definition Problem . . . . .	62
4.3	Orthographical/Spelling Criteria . . . . .	64
4.4	Morphological Criteria . . . . .	65
	4.4.1 Word Inflection . . . . .	65
	4.4.2 Constituent Inflection . . . . .	65
4.5	Phonetic/prosodic criteria . . . . .	66
4.6	Syntactic criteria . . . . .	67

## Table of Contents

4.6.1	Inseparability . . . . .	67
4.6.2	Inability to Modify the Modifier . . . . .	67
4.6.3	Inability to Replace the Head with the Pronoun <i>one</i> . . . . .	68
4.7	Semantic criteria . . . . .	68
4.7.1	Permanence . . . . .	68
4.7.2	Non-compositionality . . . . .	68
4.7.3	Lexicalization . . . . .	69
<b>5</b>	<b>Cross-lingual Observations of Compounds</b>	<b>71</b>
5.1	Parallel Compounding . . . . .	71
5.1.1	Parallel Closed Compounding . . . . .	73
5.2	Phrasal Translations . . . . .	74
5.3	Asymmetric Translations . . . . .	75
5.3.1	Aspect Alternations . . . . .	75
5.3.2	Atomic Equivalents . . . . .	76
5.3.3	Constituent Swapping . . . . .	76
5.4	Cross-lingual Indicators for Compound Analysis . . . . .	77
5.4.1	Compound Identification . . . . .	77
5.4.2	Compound Splitting . . . . .	77
5.4.3	Compound Parsing . . . . .	77
5.4.4	Prediction of Semantic Indeterminacy . . . . .	78
5.4.5	Prediction of Semantic Relations . . . . .	79
<b>6</b>	<b>Bottom Line of Nature of Compounds</b>	<b>81</b>
6.1	Summary . . . . .	81
6.2	Conclusion . . . . .	82
6.3	Motivation and Outlook . . . . .	83
	<b>Part C Compound Identification</b>	<b>85</b>
<b>7</b>	<b>Introduction to Compound Identification</b>	<b>87</b>
7.1	Motivation . . . . .	87
7.1.1	Motivation for Compound Identification . . . . .	88
7.1.2	Motivation for Multilingual Compound Resource . . . . .	88
7.2	Contributions and related Research Questions . . . . .	90
7.2.1	New Insights about the Notion of Compoundhood . . . . .	90
7.2.2	Cross-lingual Compound Identifier . . . . .	91
7.2.3	Lexical Resources . . . . .	92
7.3	Outline . . . . .	92
<b>8</b>	<b>Related Work on Compound Identification</b>	<b>93</b>
8.1	Methods for the Identification and Discovery of Compounds . . . . .	93
8.1.1	Two-Noun Compounds . . . . .	95

## Table of Contents

8.1.2	Nominal Compounds . . . . .	97
8.1.3	Bigrams including Nominal Compounds . . . . .	98
8.1.4	Closed Compounds . . . . .	98
8.1.5	General MWEs . . . . .	99
8.2	Compound Resources . . . . .	103
8.2.1	General Compounds . . . . .	105
8.2.2	Two-Noun Compounds . . . . .	107
8.2.3	Binary Nominal Compounds . . . . .	114
8.2.4	Closed Compounds . . . . .	115
8.2.5	General MWEs . . . . .	115
<b>9</b>	<b>Parallel Corpus</b>	<b>119</b>
9.1	Language Selection in Europarl . . . . .	120
9.2	Preprocessing Steps . . . . .	120
9.3	Opus Europarl . . . . .	121
<b>10</b>	<b>Pilot Studies on Compound Identification</b>	<b>123</b>
10.1	Linguistic Criterion Inspection . . . . .	123
10.1.1	Europarl Nominal Compoundhood Rating Gold Standard . . . . .	123
10.1.2	Inter-Annotator Agreement on the ENCR . . . . .	128
10.1.3	Experiments on the Linguistic Criterion Inspection . . . . .	132
10.1.4	Conclusion of the Linguistic Criterion Inspection . . . . .	139
10.2	Cross-lingual Compound Inspection . . . . .	142
10.2.1	Compound Resource . . . . .	142
10.2.2	Frequency Distributions for Aligned USPs . . . . .	142
10.2.3	Conclusion of the Cross-lingual Compound Inspection . . . . .	145
<b>11</b>	<b>Compound Identification Method</b>	<b>147</b>
11.1	Compound Candidate Selection . . . . .	147
11.2	Multilingual Complementation . . . . .	149
11.3	Cross-lingual Validation . . . . .	151
<b>12</b>	<b>Europarl Nominal Compound Database</b>	<b>153</b>
12.1	Statistics and Cross-lingual Observations in the ENCD . . . . .	153
12.1.1	PoS pattern Distribution . . . . .	153
12.1.2	Degree of Closed Compounding across Languages . . . . .	156
12.1.3	Paraphrasing and Bracketing 3NCs . . . . .	156
12.2	Additional Information . . . . .	158
<b>13</b>	<b>Experiments on Compound Identification</b>	<b>159</b>
13.1	Setup . . . . .	159
13.2	Results and Discussion . . . . .	160
<b>14</b>	<b>Bottom Line of Compound Identification</b>	<b>163</b>
14.1	Summary . . . . .	163

14.2 Conclusion . . . . .	164
14.3 Future Work . . . . .	166
<b>Part D Compound Splitting</b>	<b>169</b>
<b>15 Introduction to Compound Splitting</b>	<b>171</b>
15.1 Motivation . . . . .	171
15.1.1 The Common Statistical Approach . . . . .	171
15.1.2 Limitations of the Common Statistical Approach . . . . .	172
15.2 Contributions and related Research Questions . . . . .	175
15.2.1 Multilingual Compound Splitter . . . . .	175
15.2.2 Semantics-driven Re-ranker for Compound Splitters . . . . .	176
15.2.3 Additional Evaluation Methods . . . . .	177
15.3 Outline . . . . .	178
<b>16 Related Work on Compound Splitting</b>	<b>179</b>
16.1 Statistical Approaches . . . . .	181
16.1.1 Frequency-based Approaches . . . . .	181
16.1.2 Approaches based on Cross-lingual Information . . . . .	185
16.1.3 Approaches based on Distributional Semantics . . . . .	186
16.1.4 Approaches based on Supervised Machine Learning . . . . .	187
16.2 Linguistic Approaches . . . . .	189
16.2.1 Approaches based on a Morphological Analyzer . . . . .	189
16.2.2 Approaches based on Hand-crafted Transformation Rules . . . . .	191
16.3 Performance of Splitting Approaches . . . . .	192
16.3.1 Statistical vs. Linguistic Approaches . . . . .	192
16.3.2 Comparison in this Thesis . . . . .	193
<b>17 Morphological Operation Patterns</b>	<b>195</b>
17.1 Compilation of MOPs . . . . .	195
17.2 Sources for MOPs . . . . .	196
17.2.1 Word MOPs . . . . .	196
17.2.2 Gold-constituent MOPs . . . . .	196
17.2.3 Hand-crafted constituent MOPs . . . . .	197
17.3 MOP Application . . . . .	197
17.4 Inverting MOPs . . . . .	198
<b>18 Multilingual Compound Splitting</b>	<b>199</b>
18.1 Binary Splitter . . . . .	200
18.1.1 Split Point Markers . . . . .	200
18.2 Constituent Normalization . . . . .	201
18.2.1 Ngram Index Lookup . . . . .	201
18.2.2 MOP Application . . . . .	202

## Table of Contents

18.3	Best Split Determination . . . . .	204
18.4	Additional Compound Splitting Features . . . . .	205
18.4.1	Prior MOP Lemmatization . . . . .	205
18.4.2	Compound Content Word Restriction . . . . .	206
18.4.3	PoS Agreement Restriction for the Modifier . . . . .	207
18.4.4	Lexeme Agreement Restriction for the Head . . . . .	208
18.4.5	Trade-off between PoS and Lexeme Agreement . . . . .	209
18.5	Compound Splitting Representation . . . . .	209
18.5.1	Split Tree . . . . .	210
18.5.2	Linear Lemma Sequence Format . . . . .	211
18.5.3	Linear Split Point Format . . . . .	212
18.6	Experiments . . . . .	212
18.6.1	Target Languages . . . . .	212
18.6.2	Training Data . . . . .	213
18.6.3	Evaluation Format . . . . .	214
18.6.4	Gold Standards . . . . .	215
18.6.5	Intrinsic Evaluation Method . . . . .	223
18.6.6	External Compound Splitting Methods in Comparison . . . . .	225
18.6.7	Results and Discussion . . . . .	227
<b>19</b>	<b>Semantically Informed Compound Splitting using Shallow Semantics</b>	<b>239</b>
19.1	Distributional Semantics . . . . .	240
19.1.1	Introduction . . . . .	240
19.1.2	Formal Description . . . . .	240
19.2	Distributional Semantics for Compound Splitting . . . . .	241
19.3	Re-ranking Method . . . . .	242
19.3.1	Initial Split Ranking . . . . .	242
19.3.2	Determination of the Distributional Similarities . . . . .	242
19.3.3	Distributional Similarity Modes . . . . .	243
19.3.4	Split Score Product and Re-ranking . . . . .	244
19.3.5	Data Sparsity Treatment . . . . .	245
19.3.6	Non-compositional Compounds . . . . .	245
19.4	Experiments . . . . .	246
19.4.1	Languages . . . . .	246
19.4.2	Training Corpus . . . . .	246
19.4.3	Evaluation Measurement . . . . .	246
19.4.4	Gold Standard . . . . .	247
19.4.5	Utilized Distributional Semantics Model . . . . .	247
19.4.6	Rankings in Comparison . . . . .	247
19.4.7	Inspected Compound Splitters . . . . .	247
19.4.8	Results and Discussion . . . . .	249
<b>20</b>	<b>Extrinsic Evaluation of Compound Splitting using Recognizing Textual Entailment</b>	<b>255</b>



## Table of Contents

20.1	Introduction . . . . .	255
20.1.1	Textual Entailment . . . . .	255
20.1.2	The Benefits of RTE for various NLP Tasks . . . . .	256
20.1.3	The Lexical Overlap Approach . . . . .	256
20.1.4	Outline of this Chapter . . . . .	256
20.2	RTE and Compound Splitting . . . . .	257
20.2.1	Limitation due to Opacity of Closed Compounds . . . . .	257
20.2.2	Enriching RTE with Compound Splitting . . . . .	258
20.2.3	RTE as Extrinsic Evaluation Method . . . . .	259
20.3	Extrinsic Evaluation: SMT vs. RTE . . . . .	261
20.4	Multi-level Alignment Framework . . . . .	262
20.5	Experiment . . . . .	264
20.5.1	Simplified Algorithm . . . . .	264
20.5.2	Training and Test Set . . . . .	264
20.5.3	Supervised Classification . . . . .	265
20.5.4	RTE Evaluation Measurements . . . . .	265
20.5.5	Target Languages . . . . .	266
20.5.6	Inspected Compound Splitters . . . . .	266
20.5.7	True Casing of Constituent Lemmas . . . . .	266
20.5.8	Adding Compounding vs. Derivational Information . . . . .	267
20.6	Results . . . . .	267
20.6.1	Observations . . . . .	268
20.6.2	Discussion . . . . .	268
<b>21</b>	<b>Bottom Line of Compound Splitting</b>	<b>271</b>
21.1	Summary . . . . .	271
21.2	Conclusion . . . . .	275
21.3	Future Work . . . . .	285
21.3.1	Multilingual Compound Splitting . . . . .	285
21.3.2	Shallow Semantics Support . . . . .	287
21.3.3	Evaluation Method . . . . .	287
<b>Part E</b>	<b>Compound Parsing</b>	<b>291</b>
<b>22</b>	<b>Introduction to Compound Parsing</b>	<b>293</b>
22.1	Motivation . . . . .	294
22.1.1	The Importance of Compound Parsing . . . . .	294
22.1.2	Behaghel’s First Law - our Guiding Principle . . . . .	294
22.2	Contributions and related Research Questions . . . . .	296
22.2.1	Spatial Proximity for Semantic Association . . . . .	296
22.2.2	Cross-lingual Perspective for Token-based Parsing . . . . .	297
22.2.3	Simple Metric for Cross-lingual Spatial Proximity . . . . .	297
22.2.4	Automatic Detection of Semantic Indeterminacy . . . . .	297

22.3 Outline . . . . .	298
<b>23 Related Work on Compound Parsing</b>	<b>299</b>
23.1 Basic Approaches to Compound Parsing . . . . .	300
23.1.1 Adjacency Model . . . . .	301
23.1.2 Dependency Model . . . . .	301
23.1.3 Hybrid Adjacency-Dependency Model . . . . .	303
23.2 Association Measures . . . . .	303
23.3 Parsing of Noun Compounds . . . . .	305
23.4 Parsing of Base NPs . . . . .	312
23.5 Cross-lingual Disambiguation of other Structures . . . . .	321
<b>24 Pilot Study using Aligned Phrase Patterns</b>	<b>325</b>
24.1 Aligned Phrase Patterns . . . . .	325
24.1.1 Function of Aligned Phrase Patterns . . . . .	325
24.1.2 Manual Definition of Aligned Phrase Patterns . . . . .	326
24.1.3 Structure Class Assignment . . . . .	327
24.2 Aligned Phrase Pattern Parsing . . . . .	328
24.3 Aligned Phrase Pattern Parsing with Word Alignment Support . . . . .	329
24.4 Experiment . . . . .	331
24.4.1 Dataset . . . . .	331
24.4.2 Gold Standard Annotation . . . . .	331
24.4.3 Methods in Comparison . . . . .	331
24.4.4 Evaluation Measure . . . . .	332
24.4.5 Results . . . . .	332
24.4.6 Discussion . . . . .	333
<b>25 Compound Parsing Methods using Aligned Word Distance</b>	<b>335</b>
25.1 Aligned Word Distance . . . . .	335
25.2 Deterministic Bottom-Up Parsing . . . . .	337
25.2.1 The Algorithm . . . . .	337
25.2.2 Example cases . . . . .	339
25.2.3 Experiments . . . . .	340
25.2.4 Discussion and Conclusion . . . . .	345
25.3 Non-deterministic Tree Accumulation Parsing . . . . .	346
25.3.1 Principle of a Semantically Valid Parse Tree . . . . .	347
25.3.2 Non-deterministic Full Tree Accumulation Parsing . . . . .	348
25.3.3 Non-deterministic Subtree Accumulation Parsing . . . . .	353
25.3.4 Experiments . . . . .	358
<b>26 Bottom Line of the Compound Parsing</b>	<b>363</b>
26.1 Summary . . . . .	363
26.2 Conclusion . . . . .	365
26.3 Future Work . . . . .	369

Table of Contents

26.3.1	Parsing with Bootstrapped Aligned Phrase Pattern Set . . . . .	369
26.3.2	Weighted Aligned Word Distance . . . . .	371
26.3.3	Monolingual Word Distance Metric for Compound Parsing . . . . .	373
26.3.4	Hybrid Adjacency-Dependency Model . . . . .	374
26.3.5	Revised Dependency Model . . . . .	377
26.3.6	Evaluation Setup for Illustrating the Potential of Token-based Compound Parsing . . . . .	377
26.3.7	Adaptation of Cross-lingual Metric-based Compound Parsing on Non-parallel Data . . . . .	378
26.3.8	Exploiting Cross-lingual Supervision for Monolingual Training . . .	379
<b>Part F Bottom Line</b>		<b>381</b>
<b>27 Summary, Conclusion and Future Work of the Thesis</b>		<b>383</b>
27.1	Summary of the Thesis . . . . .	383
27.2	Conclusion of the Thesis . . . . .	385
27.3	Future Work of the Thesis . . . . .	388
<b>Part G Appendix</b>		<b>391</b>
<b>A Universal Surface Patterns</b>		<b>393</b>
A.1	Motivation . . . . .	393
A.1.1	Language Independence . . . . .	393
A.1.2	Complexity of Nouns . . . . .	393
A.1.3	Functional Context . . . . .	393
A.2	Transformation of PoS Patterns to USPs . . . . .	394
A.2.1	Simplified USPs . . . . .	394
<b>B German Constituent Inflection</b>		<b>395</b>
<b>C Split Point Format Compilation</b>		<b>397</b>
C.1	Linear Compilation of the Split Point Format . . . . .	397
C.2	Hierarchical Compilation of the SPF . . . . .	398
C.2.1	SPF by using MOP Application . . . . .	399
C.2.2	SPF by using Linear Approach for Constituent Forms . . . . .	400
C.3	Experiments . . . . .	400
C.3.1	Linear SPF Compilation . . . . .	400
C.3.2	Hierarchical SPF Compilation . . . . .	401
C.4	SPF Compilations of Resources . . . . .	403
C.4.1	SPF for HH2011GS . . . . .	403
C.4.2	SPF for M2006GS . . . . .	403
C.4.3	SPF for FF2010 . . . . .	403

*Table of Contents*

<b>D Further Compound Splitting Gold Standards</b>	<b>405</b>
D.1 German Splitting Gold Standard of Cap (2014) . . . . .	405
D.2 Ghost-NN . . . . .	406
<b>E Annotation Guidelines for Creating the Europarl Nominal Compound- hood Ratings</b>	<b>407</b>
E.1 Introduction . . . . .	407
E.2 Rating of Linguistic Criteria for Compoundhood . . . . .	408
E.3 Please note . . . . .	410
E.4 Annotation process . . . . .	411
E.5 Training and annotation stage . . . . .	412
<b>List of Abbreviations</b>	<b>413</b>
<b>List of Algorithms</b>	<b>421</b>
<b>List of Figures</b>	<b>425</b>
<b>List of Publications</b>	<b>429</b>
<b>List of Tables</b>	<b>431</b>
<b>List of Terms</b>	<b>435</b>
<b>Bibliography</b>	<b>457</b>

Part A.

Preface



# 1. Introduction to the Thesis

- SIE: Ein Nashorn.- Du Benedigt, warum heisst denn dös ‘Nashorn’?  
(A ‘Nashorn’ (rhinoceros) - Benedict, why is this called a ‘Nashorn’?)
- ER: Sehr einfach - weil’s auf der Nase ein Horn hat.  
(That’s obvious - it has a ‘Horn’ (horn) on its ‘Nase’ (nose).)
- SIE: Ja, wie is’ dann das beim Elefant?  
(What about ‘Elefant’ (elephant)?)
- ER: Der hat eine ‘Ele’ am ‘Fant’.  
(That one has an ‘Ele’ on its ‘Fant’.)
- SIE: Nein, der hat einen Rüssel am Kopf - der müsste eigentlich  
Rüsselkopf heissen.  
(No, it has a ‘Rüssel’ (trunk) on its ‘Kopf’ (head) - it should be  
called ‘Rüsselkopf’ (trunk head))

(Valentin and Bachmaier, 1995, p. 153)

Lexical units (or [lexemes](#)) are the building blocks of natural language. A common starting point of many [Natural Language Processing \(NLP\)](#) tasks is the determination and [understanding](#) of [lexemes](#), which already poses big challenges. Besides [atomic lexemes](#) (i.e., [lexemes](#) that cannot be broken down into several [content lexemes](#)), natural language is full of [complex lexemes](#) (i.e., [lexemes](#) that are composed of several [atomic lexemes](#)). [Complex lexemes](#) are an important subject of study in theoretical linguistics, because they constitute a continuum from fully compositional (e.g., *apple pie*) to idiosyncratic (e.g., *honeymoon*) [word](#) formations and are found at the boundary between ‘[words](#)’ (e.g., *French toast*) and phrases (e.g., *French car*). The major representative of the class of [complex lexemes](#) is the [compound](#) (e.g., *network*, *handbag*, *paper-clip* or *natural language processing*). As defined by Bauer (2003), a [compound](#) is “the formation of a new [lexeme](#) by adjoining two or more [lexemes](#)”. [Compounds](#) are a lexical phenomenon which is abundant in many languages and occur in various embodiments. For example, most Germanic languages are so-called [closed compounding languages](#), i.e.,

languages that realize **compounds** as one-word constructions, the so-called **closed compounds** (e.g., the German *Dampfschiffahrt* ‘steam navigation’). A semi-closed variant is the **hyphenated compound** (e.g., the Dutch *auto-industrie* ‘automotive market’).

Despite the prevalence of **compounds** in many languages, the definition of **compounds** and even their existence has been controversially discussed in linguistics literature. We will outline this discussion in Chapter 4. While most previous NLP methods dealing with **compounds** focused on commonly non-debatable cases of **nominal compounds** (e.g., sequences of two English nouns, i.e., **two-Noun Compounds (2NCs)**), we think that addressing the complex and controversial issue of **compoundhood** definition is most challenging.

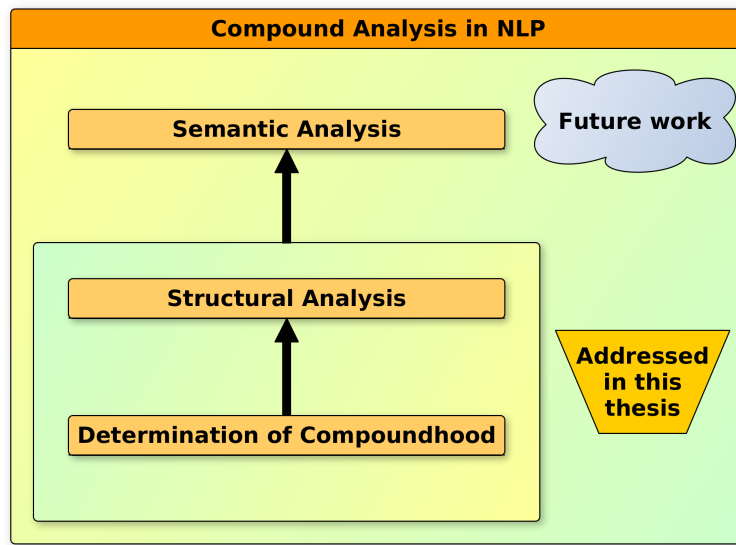


Figure 1.1.: Compound Analysis in NLP

For the ultimate goal of **understanding** the meaning of **compounds** (i.e., its semantic analysis), it is inevitable to first conceive the **notion of compoundhood** and the distinction from similar constructions such as phrases. After having a clearer picture of what defines **compounds** and having determined the **compoundhood status** of an expression  $\Psi$ , the subsequent step towards **understanding**  $\Psi$  concerns the **structural analysis**, which aims to answer questions about the **internal structure** of  $\Psi$ , i.e., what are the **immediate** and mediate **constituents** of  $\Psi$ . The **compound identification** and the **structural analysis**, addressed in this thesis, serve as basis for the semantic analysis, which we leave for future work, as illustrated in Figure 1.1.



## 1.1. Thematic Structure of the Thesis

The work presented in this thesis can be divided into two areas as illustrated in Figure 1.2.

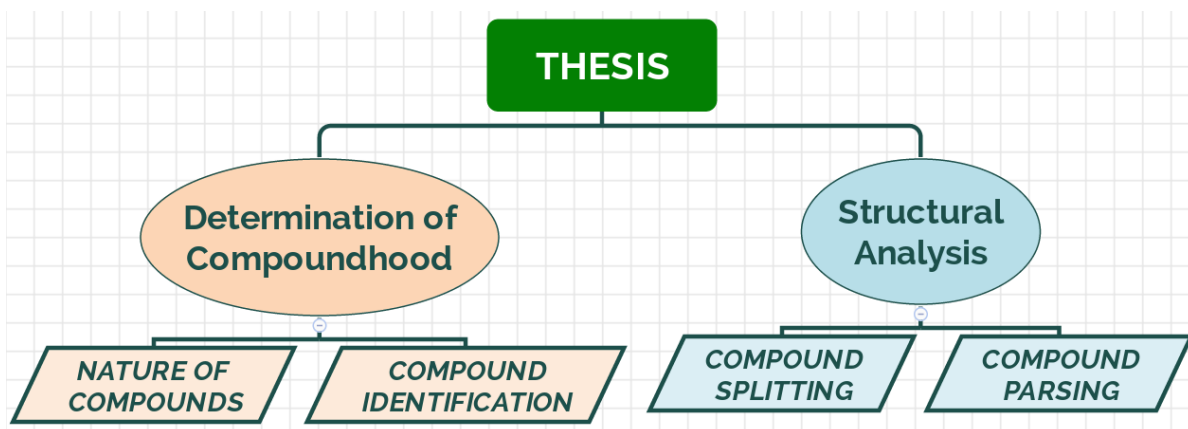


Figure 1.2.: Thematic structure of the thesis

1. **Determination of Compoundhood:** In this area, we research the characteristics of *compounds*, i.e., we provide background information on the **nature of compounds** and propose a **compound description** inspired by **Linguistic Criterion Inspection**. Various *linguistic criteria* for *compoundhood*, proposed in literature, are inspected and their validity for the determination of *compoundhood* is rated. We observed some *cross-lingual* regularities on the spelling forms of *equivalents* in **Cross-lingual Compound Inspection**. These observations guide us for designing a *cross-lingual compound identification method*.

The area of **compoundhood determination** is the basis for the second area.

2. **Structural Analysis of Compounds:** In this area, we present methods for two sub-tasks of automatic *structural analysis* (i.e., determining the *internal structure*) of *compounds*.

Firstly, we address the task of **compound splitting**, i.e., determining the composed *lexemes* of a *compound*. While this task is trivial for *open* or *hyphenated compounds* (i.e., a simple tokenization at whitespaces or hyphens), for opaque *closed compounds*, which are the common *target*, an advanced **compound splitter** is necessary. For example, the German *three-Noun Compound* (3NC) *Hühnersuppenrezept* ‘chicken soup recipe’ has to be *split* into the *constituent lemmas*

## 1. Introduction to the Thesis

*Huhn* ‘chicken’, *Suppe* ‘soup’ and *Rezept* ‘recipe’. For determining the lemma *Huhn*, the constituent form *Hühner* has to be morphologically normalized using various operations (e.g., reducing the Umlaut *ü* to *u* and truncating the *er*-suffix), as illustrated in Figure 1.3.

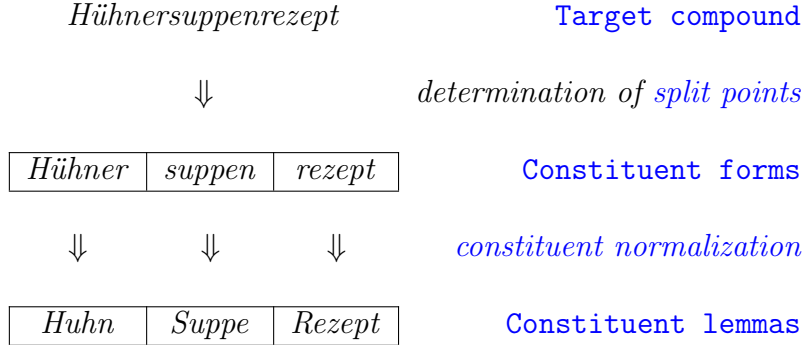


Figure 1.3.: Linear splitting for *Hühnersuppenrezept*

As will be discussed in the course of this thesis, these **normalization** operations are non-trivial and vary from **lexeme** to **lexeme**. For example, while the **modifier lemma** of the German 2NC *Rinden*|*mulch* ‘bark mulch’ is *Rinde* ‘bark’, the **modifier lemma** of the 2NC *Rinder*|*zucht* ‘cattle breeding’ is *Rind* ‘cattle’.

Besides determining the meaning of the composed (**atomic**) **lexemes**, the ultimate goal (i.e., the semantic analysis) also includes the determination of the underlying **semantic relation** holding between (**immediate**) **modifier** and **head** (which is outside the scope of this thesis). Thus, for **compounds** having three or more **atomic constituents**, we first have to figure out which **constituents** can be grouped together, forming the **immediate constituents**. Therefore, in the second part of the **structural analysis**, we deal with the task of **compound parsing** (also called **bracketing**), targeting complex **open compounds** such as *natural language processing* to be commonly analyzed as  $[[\textit{natural language}] \textit{processing}]$ . For our above example, the split 3NC *Huhn*|*Suppe*|*Rezept*, we need to know whether *Huhn* has to be grouped with *Suppe*, or *Suppe* has to be grouped with *Rezept*, leading to two different **parse trees** as shown in Figure 1.4, where the LEFT-branching interpretation seems more plausible (i.e., the recipe for a chicken soup), where the underlying **semantic relation** might be denoted as ABOUT.

Figure 1.4.: Split tree for *Hühnersuppenrezept*

## 1.2. Motivation of the Thesis

The motivation for this thesis is divided into two parts. The first part (1.2.1) concerns the subject of our research, **nominal compounds**. Why did we choose to research **nominal compounds** and what is the challenge in processing them? The second part (1.2.2) describes the motivation for the choice of our methods and experiments on **compound analysis** that will be elaborated in the course of this thesis.

### 1.2.1. Motivation for Analyzing Compounds

#### Productivity of **Compounds**

The major class of **complex lexemes** are **compounds**, and within this group, **nominal compounds**, i.e., **compounds** having a nominal head. **Nominal compounds** are a very productive **word formation type** for language producers at any age. Even 2-year-olds can understand and 4-year-olds can produce new **lexemes** by using **compounds** consisting of two **atomic lexemes** without mistakes (Clark, 1981). Language users can both produce and understand unknown **compounds** without much cognitive effort. “Almost any pair of English nouns can be combined to produce a valid, if not always sensible, **compound**” (Ó Séaghdha, 2008). Thus, even **compounds** based on commonly unrelated **constituents** can be processed by humans. For example, a *chocolate window* can get the meaning of a brown-colour window glass or it can be understood as a *shop window* displaying some chocolate (cf. deictic **compounds** (Downing, 1977)). The high productivity of **compounds** can also be observed when inspecting text corpora. For example, analyzing the German APA corpus, Baroni et al. (2002) observed that almost half (47%) of the **word types** were **compounds**. In fact, **compounding** is indispensable in our everyday life. Consider the following scenery which is packed full of **nominal compounds**: in the *late-*

*night*, a man enters his *living room*, steps on a *designer carpet*, the *remote control* lies on the *coffee table*, next to a bag of *potato chips*, his *Persian cat* sleeps in the *armchair*, the *wallpaper* shows a *floral pattern*, his *Digital Versatile Disc (DVD) player* connected to his *high definition television* plays the *action movie, Cliffhanger* with the *Academy Award winner* Sylvester Stallone. The strong productivity of **compounds** motivates us to research the nature of **compounds** and develop methods for **compound analysis**.

### Type-Token Ratio of Compounds

While the frequency of any **compound types** in natural language text is very high (in many languages), most **compounds** (in particular non-lexicalized constructions) occur with a very low **token** frequency. As mentioned above, in the German APA corpus almost half (47%) of the **word types** were **compounds**. In contrast, **compounds** accounted for a small portion of the overall **token** count (7%). Many **compounds** were rare (83% of the **compounds** had a corpus frequency of 5 or lower). This characteristics of **compounds**, i.e., being abundant as a phenomenon but scarce in terms of individual examples, makes their analysis particularly problematic for statistical techniques that require high **token** frequencies for making accurate predictions. Data sparsity of **compounds** is expected to lead to low performance when treating **compounds** as opaque **lexemes**. Listing all possible **compounds** (with all necessary attributes) in a lexical resource would be as infeasible as listing all possible adjective-noun combinations. A more detailed discussion about the productivity and frequency distribution of **compounds** is given in Section 3.3.

As a consequence, previous **NLP** attempts for the automatic analysis of **compounds** proposed compositional approaches, i.e., the analysis of **compounds** is based on the analysis of its **constituents**. This kind of analysis usually expects a **target compound** whose meaning is based on the meaning of its **constituents** (cf. semantic compositionality, Section 3.8.1).

The **type-token** ratio of **compounds** motivates us to investigate **compoundhood** and develop new compositional methods for **compound analysis**.

### High Degree of Ambiguity of various Compound Analysis Levels

The automatic analysis of **compounds** poses a big challenge for various uncertainties on different levels of analysis.

On the ground level (i.e., the determination of **compoundhood**), we have to deal with the **ambiguity of the compoundhood status**. For example, due to the ambiguity of derivational suffixes, the **lexeme** *friendship* can be interpreted as **atomic** (derived) **word**

or as a **2NC** *friend* | *ship* (e.g., a ship shared by a certain group of friends). We can observe this type of ambiguity in various languages. For example, the German **word** *Instrumentarien* can be interpreted as the plural form of the **atomic** *Instrumentarium* ‘apparatus’ or as the pluralized **2NC** *Instrument* | *Arien* ‘instrumental arias’. Another type of ambiguity on this level concerns the distinction between a phrasal and a **compound** interpretation. For example, the **adjective-noun** sequence *French teacher* can be interpreted as a phrase (i.e., a teacher having a French nationality) or as **nominal compound** (i.e., a person teaching the school subject ‘French’).

On the next level, we have to deal with **structural ambiguity**. For example, the uncertainty about the position of the **split point** for **closed compounds** (e.g., the German *Gastraum* can be split into *Gas* | *Traum* ‘gas dream’ or *Gast* | *Raum* ‘guest room’). For **compounds** having three or more **constituents**, we have to resolve ambiguity about the syntactic structure (e.g.,  $[[\textit{natural language}] \textit{processing}]$  vs.  $[\textit{natural} [\textit{language processing}]]$ ).

Finally, on the abstract (semantic) level, there are various types of **semantic ambiguity**. First of all, we need to determine the meaning of all (**atomic**) **constituents** (possibly requiring **Word Sense Disambiguation (WSD)**). Next, based on the **constituents’** meaning, we have to determine the degree of compositionality (e.g., *honeymoon* vs. *orange peel*). For fully non-compositional **compounds**, there is no need for a compositional interpretation. In contrast, for compositional **compounds**, we have to uncover the implicit **semantic relation** that holds between the (**immediate**) **constituents**. There are virtually infinite possibilities for interpreting these **semantic relations**. For example, the intended meanings of the following knives are based on different **semantic relations**: *cheese knife* (**object of cutting event**), *pocket knife* (**storage location**), *[stainless steel] knife* (**material**) or *hunting knife* (**purpose**). Even when knowing the **constituents’** semantic classes, the implicit **semantic relation** remains ambiguous. For example, **substance + vessel**: while *paper tray* describes a content relation (i.e., a *tray* that contains *paper*), a *plastic tray* has the intended meaning based on a material relation (i.e., a *tray* made out of *plastic*). Furthermore, the determination of the underlying **semantic relation** can depend on context. For example, the German **2NC** *Babybauch* (lit: ‘baby + belly’) can be interpreted as *baby belly* (i.e., the *belly* of a *baby*) or as *pregnant belly* (i.e., the (bigger) *belly* due to a *baby*). A semantic **compound** feature similar to the **semantic relation** is the **compound class** (e.g., whether a **compound** includes a **constituent** denoting the semantic **head** (**endocentric** e.g., *sun glasses*) or not (**exocentric** e.g., *cutthroat*)). Figure 1.5 summarizes the discussed ambiguities on all **compound analysis** levels.

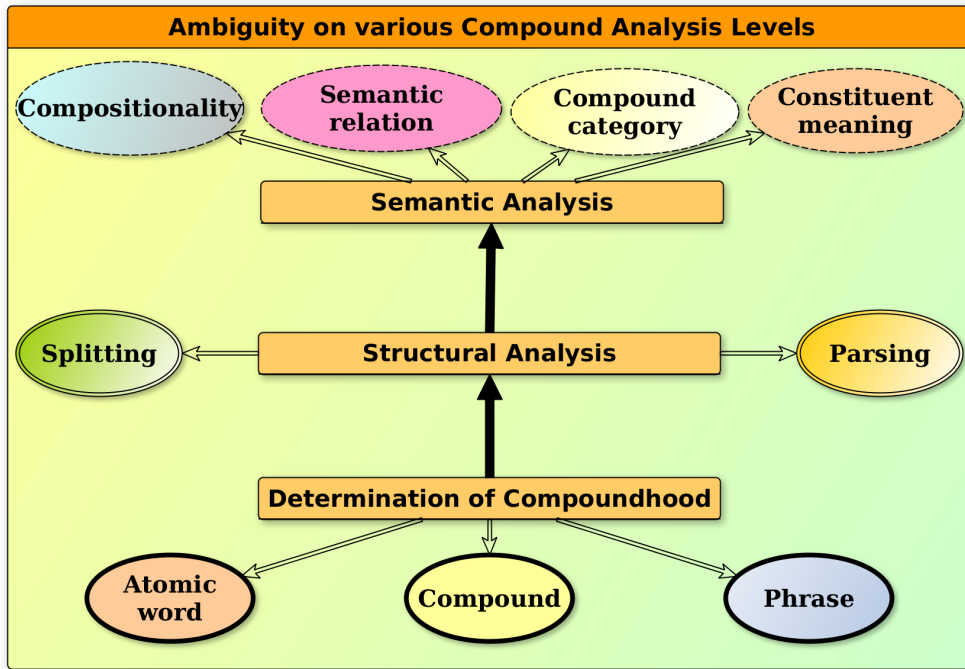


Figure 1.5.: Ambiguity in various Compound Analysis Levels

While language users commonly consult their world knowledge for resolving the ambiguity on all levels, automatic **compound analysis** is a non-trivial task. We hope that the insights in **compounding** and the analysis methods provided along with this thesis can contribute to future research on **compounding** and on the automatic analysis of **compounds**.

### Compounds as a Unique Linguistic Phenomenon

**Compounds** are an important subject of study in **theoretical linguistics**. One reason for this is that **compounds** constitute a continuum from a fully compositional to an idiosyncratic **word** formation and that **compounds** are found at the interface between **words** (lexicon) and phrases (syntax) (Ziering and Van der Plas, 2014). While the German adjective-noun **compound** *Altpapier* ‘recovered paper’ has partly lost its phrasal function (i.e., *altes Papier* ‘old paper’), for German **compounds** such as *Optimallösung* ‘optimal solution’, there is no functional difference to the corresponding phrase (i.e., *optimale Lösung* ‘optimal solution’) (Schlücker and Hüning, 2010).

Another interesting phenomenon concerns **synthetic compounds**, i.e., **noun compounds** having a deverbal **head**. According to Grimshaw (1990), deverbal nouns are ambiguous

between an argument-supporting nominal (ASN) reading (i.e., verbal arguments are inherited, as in *the assignment of the tasks*) and a Result Nominal (RN) reading (i.e., the noun has lost its subcategorical function, as in *a two-page assignment*). Iordachioaia et al. (2016) performed some experiments proving their hypothesis saying that **synthetic compounds** having a **head** with an ASN reading commonly realize the syntactic object as **modifier**, whereas **synthetic compounds** with an RN reading allow for many other interpretations, similar to root compounds.

By providing appropriate and correct analyses of **compounds**, we aim to support theoretical linguistics research.

### Importance of Compound Analysis for NLP

A correct analysis of **compounds** is inevitable for many **NLP** tasks, more specifically tasks depending on **Natural Language Understanding (NLU)**.

An **NLU**-dependent task for which **compound analysis** was deemed relevant in previous work is **Machine Translation (MT)** (Bouillon et al., 1992, Johnston and Busa, 1999, Navigli et al., 2003, Rackow et al., 1992). The accurate translation of **compounds** is non-trivial, because we find a large amount of variation and varying degrees of explicitness in the way languages deal with **compounding**, e.g., **closed compounding** in Germanic languages such as Dutch (e.g., *doodstraf* ‘death penalty’), whereas in Romance languages, such as French, **compounding** is a very infrequent **word** formation type. Most nominal **complex lexemes** are realized with postmodifying **Prepositional Phrases (PPs)** (i.e., **complex nominals**, e.g., *peine de mort* (lit: ‘penalty of death’)) or adjectives (*peine capitale* (lit: ‘penalty capital<sub>ADJ</sub>’)).

For a **Text-to-Speech (TTS)** system, it is important to know whether a **word** sequence constitutes a **compound**. For example, giving the expression *French teacher* a primary stress on the first element (as done for **compounds**) denotes a person teaching French, whereas an equal stress or a primary stress on the second **word** (as done for phrases) denotes a teacher having a French nationality (Levi, 1978).

For **Recognizing Textual Entailment (RTE)**, the knowledge about the **internal structure** of a **compound** can help to decide whether a hypothesis follows from a text. For example, for recognizing an **TE** between the text ‘*Peter has a housewife*’ and the hypothesis ‘*Peter has a wife*’, we have to perform **compound splitting** to determine the **constituents** *house* and *wife* (and ideally a semantic analysis revealing the compositional character of *housewife*). Experiments on **RTE** and **compound splitting** (Jagfeld et al., 2017), that will be discussed in more detail in Chapter 20, have shown the usefulness of

compound splitting for RTE.

Other NLP tasks that require information about the structure and meaning of compounds include Question Answering (QA), Information Extraction (IE) and Information Retrieval (IR). Nakov (2013) provided some medical examples of compound information for NLP tasks: a QA system has to know whether *tumor suppressor protein* can be interpreted as *protein acting as a tumor suppressor*; an IE system has to decide whether the compounds *neck thrombosis* and *neck vein thrombosis* are co-referent. In an IR system, a query containing the compound *migraine treatment* can be expanded with verbs like *relieve* or *prevent* for optimizing the retrieval results (Nakov, 2013).

## 1.2.2. Motivation for our Methodology

### Linguistics Background

Most previous linguistically motivated work relies on linguistic resources such as grammars or lexicons. While our work is based on linguistics background, we rely on linguistics theories and regularities for the assumptions underlying our approaches but avoid the use of manual resources such as grammars or lexicons. We describe some in more detail below.

In order to determine the compoundhood status of an expression  $\Psi$ , we consider various linguistic criteria for compoundhood described in linguistics literature (Lieber and Štekauer, 2009, Nakov, 2013). These criteria cover most areas of linguistics: orthography (4.3), morphology (4.4), phonetics and prosody (4.5), syntax (4.6) and semantics (4.7).

On the morphology of compounds, we consider some regularities in the analogy between linking elements and suffixation in regular word inflection, described in different linguistic theories (e.g., Neef (2009)). These regularities serve as basis for the compound splitting task.

The linguist Otto Behaghel (1854-1936) described a set of universally valid linguistic laws about the position of words and phrases within a sentence. A very important contribution of Behaghel (1909) is his First Law saying that words or phrases that belong close together intellectually (i.e., those that have a strong semantic association) are also positioned close together. We use semantic association approximated by exploiting differences in the sentence positions across languages, serving for the compound parsing task.



## Avoidance of using Manual Resources

As described above, the methods for [compound analysis](#) presented in this thesis are linguistically motivated. However, instead of using large amounts of manual resources (e.g., hand-crafted rules or lexical resources), we aim to use as little manual support as possible for being as flexible and language-independent as possible.

For example, a major drawback of rule-based [compound analysis](#) tools is that the knowledge bottleneck of directly supervised methods (i.e., the dependence on domain and language experts and the limitation to a [target language](#)) is shifted from training data annotation to the hard-coded rule design. For example, Fritzing and Fraser (2010) developed a [compound splitting](#) method for German [closed compounds](#) which heavily relies on a manual resource, i.e., the rule-based morphological analyzer SMOR (Schmid et al., 2004). Although the approach of Fritzing and Fraser (2010) provided precise analyses for German [compounds](#) with common [constituents](#), it cannot be easily adapted to other [target languages](#) and novel [constituents](#) cannot be found until they are part of SMOR.

Another category of manually supported approaches relies on lexical resources such as WordNet (Miller, 1995a) or bilingual dictionaries (e.g., dict.cc). There are several issues in using such resources. Firstly, they are often designed for a specific domain or language, secondly lexical resources often struggle with coverage, and finally, they are based on human annotations (and as such underlie the bias of annotators' individual intuitions like annotated training data).

Besides a few simple and language-independent rules that will be described in the subsequent chapters, the main resource used in our [compound analysis](#) methods is a [parallel corpus](#).

As discussed above, there is a great benefit of using [cross-lingual supervision](#) in NLP. However, we are aware of the fact that [parallel corpora](#) are sparse and often domain-specific. Ideally, we would like to exploit the benefits of [cross-lingual supervision](#) without being restricted by data sparsity and domain specificity.

As will be shown in Part D, for the task of [multilingual compound splitting](#), we use another type of [indirect supervision](#), viz. [supervision based on morphological regularities](#), allowing for eschewing the usage of [parallel data](#) (as has been done by Macherey et al. (2011)).

Nonetheless, there is a versatile range of applications for [parallel data](#), and due to the progress of technical globalization, we expect that [parallel corpora](#) will become abundant and the issue of data sparsity will be mitigated in the future.

## Indirect Supervision

**Issues of Direct Supervision.** Most methods in statistical NLP (in particular those based on machine learning) can be categorized into **directly supervised** (or **semi-supervised**) and **unsupervised** approaches. The distinctive factor for this categorization is the need for training data which is manually annotated with possible output variables for the NLP task at hand. For example, **words** labelled with **Part-of-Speech (PoS)** tags serve for the training of a directly supervised **PoS** tagger. Unsupervised methods do not rely on manually annotated training data. Instead, such approaches exploit regularities in natural language. For example, inspired by the distributional hypothesis (Harris, 1954), saying that **words** occurring in similar contexts have a similar meaning, the **Distributional Semantics (DS)** exploit the distribution of **words** for modelling **word** meaning without the need for manual annotation of **word** meaning.

One major drawback of directly supervised methods is the dependence on manual annotation (Vlachos, 2011). This makes directly supervised approaches less flexible for the application to new domains or languages, because there is need for domain (or language) experts. Moreover, the creation of manually annotated training data is costly and time-consuming. Finally, the linguistic annotation by humans is generally biased by the individual intuitions about language. This bias often has a measurable impact on the quality of the training data (and thereby on the trained system): for many abstract NLP tasks (e.g., those based on semantics), the **Inter-Annotator Agreement (IAA)** is only moderate. For example, in the task of determining the **semantic relation** in **2NCs** using an inventory of 35 **semantic relations**, Girju et al. (2005) observed an **IAA** of  $\kappa = 0.58$ , meaning moderate agreement (Landis and Koch, 1977).

On the other hand, since regularities in natural language are often not as reliable as human annotations, fully unsupervised approaches usually show a worse performance than supervised approaches in many NLP tasks.

**Exploitation of Task-independent Information.** As a consequence, we decided to avoid directly supervised approaches and thus restrict the need for human annotations to the final evaluation of the developed methods.

While we mostly avoid direct labels for the **compound analysis** task at hand, it is possible to achieve comparable knowledge indirectly from task-independent information that occurs naturally in data (e.g., expressive translations in a **parallel corpus**) using a *transfer function* (e.g., a **cross-linguistic** theory) that maps this data to (indirectly obtained) labels for the underlying **compound analysis** task. For example, while creating a **parallel corpus**, human decisions (e.g., about the **compoundhood status** of a **word** sequence)

are derived indirectly and intuitively, because human translators focus on another task (i.e., producing a well-formed translation). Thus, the drawbacks of directly supervised approaches are not an issue. The indirectly obtained labels provide an approximation to a gold standard comprising direct labels.

In this thesis, we will distinguish between two types of *indirect supervision*: *cross-lingual supervision* and *supervision based on morphological regularities*.

**Cross-lingual Supervision.** As discussed above, while **compounding** is a universal phenomenon, we can see strong differences in how English **compounds** are translated to various languages, e.g., as **closed** or **hyphenated compounds**, as **open compounds**, as phrases (e.g., **complex nominals**) or even as **atomic lexicalized words**. Moreover, phrases are often translated differently than **compounds**. For example, the English phrase *French teacher* is translated to German as *französischer Lehrer* (i.e., a teacher having a French nationality), whereas the homographic **compound** *French teacher* is translated to German as *Französischlehrer* (i.e., a person teaching French). A more detailed discussion on **cross-lingual** observations of **compounding** is given in Chapter 5. In general, some natural languages express certain information more explicitly than other languages. This becomes apparent when working with **parallel data**. After recognizing and extracting this source of information from one language, it can be propagated to aligned languages and utilized in any suitable classification process as a kind of *naturally occurring supervision* (Snyder and Barzilay, 2010). The variety of **cross-lingual equivalents** of **compounds** provides a valuable natural knowledge source for many types of **compound analysis** (e.g., **compound identification** (Ziering and Van der Plas, 2014), **compound splitting** (Brown, 2002, Koehn and Knight, 2003, Macherey et al., 2011), **compound parsing** (Ziering and Van der Plas, 2015a,b) or determination of implicit **semantic relations** (Girju, 2007)). Some of these **compound analysis** tasks will be addressed **cross-lingually** in the course of this thesis. The character of **cross-lingual supervision** as a type of *indirect supervision* allows for exploiting human knowledge (e.g., a human translator creates the phrasal or compound equivalent of a **target compound**  $\Psi$  according to the way he or she conceives the intended meaning and **compoundhood status** of  $\Psi$ ) without being restricted to direct annotations with a fixed set of labels. The human decisions (e.g., about the **compoundhood status**) are derived indirectly and intuitively, because the language producer focuses on another task (e.g., producing a well-formed and semantically equivalent translation). Thus, the drawbacks of directly supervised approaches described above are not an issue. For example, there are no annotation costs, because **parallel data** (e.g., as occurring in patent claims, in the proceedings of the European Parliament or in movie

subtitles) is already available for many tasks outside of the NLP domain.

**Supervision based on Morphological Regularities.** Neef (2009) argues that German [linking elements](#) are derived from genitive and plural markers. Based on such theories, we exploit the regularities in the analogy between the morphology of [constituents](#) ([constituent inflection](#)) and [atomic words](#) ([word inflection](#)), and approximate [constituent inflection](#) by using [word inflection](#) operations.

## Context-based Analysis

Another benefit of [cross-lingual supervision](#) is that [parallel data](#) provide several [cross-lingual equivalents](#) of a [compound token](#) (i.e., of a certain instance of a [compound](#)) in context. Therefore, all our methods based on [cross-lingual supervision](#) allow for treating each [compound token](#) with its [cross-lingual equivalents](#) individually in its given context. Thereby, [cross-lingually supervised](#) methods can deal with context-dependent ambiguity, which occurs on various levels of [compound analysis](#), as illustrated in Figure 1.5 above.

## 1.3. Overview of the Main Research Questions

This section provides an overview of the main research questions that have led our research in monolingual and [cross-lingual compounding](#) and the development of new [compound analysis](#) methods.

### 1.3.1. Compoundhood

The first main research question concerns the notion of [compoundhood](#).

**RQ\_1:** What are [compounds](#)?

Both the definition and even the existence of [compounds](#) is discussed controversially in linguistics literature (Lieber and Štekauer, 2009). As key for determining the [compoundhood status](#) of an expression  $\Psi$ , most linguists propose various more or less reliable [linguistic criteria](#). From a practical perspective, we aim to find the best-working [criteria](#)

**RQ\_1-A:** What [linguistic criteria](#) help to identify [compounds](#)?

A big challenge for [cross-lingual NLP](#) methods (e.g., [Machine Translation \(MT\)](#)) is the large variety of formations of [cross-lingual equivalents](#) of a [compound](#). We aim to find the most representative ways how [compounds](#) are expressed in different languages.

**RQ\_1-B:** What are the most frequent formations of **cross-lingual equivalents** of an English **compound**?

Actually, there is **cross-lingual** evidence in terms of spelling correlating with the **compoundhood status**, e.g., *French teacher* being translated to German as *Französischlehrer* (**compound**) or *französischer Lehrer* (phrase). We aim to investigate the potential of **cross-lingual** evidence for determining the **compoundhood status** of a **target** expression.

**RQ\_1-C:** Is **cross-lingual** information beneficial for the automatic **identification** of **compounds** in context?

### 1.3.2. Indirect Supervision and Avoidance of Manual resources

The second main research question concerns **indirect supervision** on **compound analysis**, avoiding human annotations for the underlying analysis task and instead exploiting task-independent information as approximation for direct labels.

**RQ\_2:** Does the automatic **analysis of compounds** based on **indirect supervision** lead to good results?

There are various sources for **indirect supervision** such as regularities and analogies on different linguistic levels (e.g., morphology) or **cross-lingual equivalents** for a **target compound**.

**RQ\_2-A:** What sources of **indirect supervision** can we use for **compound analysis**?

Another key concept in our **compound analysis** methodology is the avoidance of manual resources such as hand-crafted rules or lexical resources (e.g., WordNet). Manual-resource-rich approaches on **compound analysis** enjoy knowledge directly tailored for the underlying analysis task, leading to a higher analysis precision.

**RQ\_2-B:** How do manual-resource-lean methods compare to resource-rich and language-specific approaches?

A crucial benefit of avoiding manual resources is the independence of language and domain.

**RQ\_2-C:** How language-independent are our **compound analysis** approaches and what resources do they still need?

## 1.4. Main Contributions of the Thesis

Besides minor observations made during data analysis and experiments (e.g., error analyses), and developed methods for [compound identification](#) and [structural analysis](#) ([compound splitting](#) and [compound parsing](#)), we claim to provide the following theoretical main contributions along with this thesis.

### 1.4.1. New Insights about the Notion of Compoundhood

The first main contribution concerns the notion of [compoundhood](#). While previous work in linguistics literature discusses the definition of [compounds](#) controversially, describes some more or less reliable [linguistic criteria](#) for [compoundhood](#) and presents counterexamples for each of them, the field of [NLP](#) mostly avoids to tackle the definition and [identification](#) of [compounds](#) but relies on commonly non-debatable cases of [nominal compounds](#) (e.g., German [closed compounds](#) or sequences of two English nouns, i.e., [binary noun compounds](#)). To the best of our knowledge, there is no previous work in computational linguistics that addresses the definition of [compounds](#) and the distinction from phrases or [atomic lexemes](#), as well as the automatic [identification](#) of any [nominal compounds](#) in general.

Our research and experiments shed new light on the notion of [compoundhood](#).

**Insights for the Definition.** We inspected various [linguistic criteria](#) for the definition of [nominal compounds](#) in a corpus study. To this end, we make use of human ratings of [linguistic criteria](#) for different kinds of [nominal compound](#) candidates. We show which of the commonly established [linguistic criteria](#) are most and least reliable for the definition of [compounds](#) in *English* (Section 10.1).

**Cross-lingual Insights.** We provide a qualitative study on the [cross-linguistic](#) behavior of [nominal compounds](#), i.e., we show cases of [parallel compounding](#), phrasal and asymmetric translations (Chapter 5), and present a quantitative study on the way how [cross-lingual equivalents](#) of English [2NCs](#) are formed (Section 10.2).

### 1.4.2. Potential of Cross-lingual Evidence on Association Strength for Compound Analysis

As discussed above, we avoid the usage of direct supervision for our methods. Instead, our methods rely on [indirect supervision](#), such as the [cross-lingual supervision](#). The second main contribution concerns the potential of manual-resource-lean features of [cross-](#)

lingual supervision for compound analysis. Previous work on compound analysis use cross-lingual evidence only as back-off feature (e.g., for the compound splitting (Brown, 2002)) or as lexical feature propagating semantics across languages (e.g., using Romance prepositions of complex nominals for the supervised learning of semantic relations of binary noun compounds (Girju, 2007)).

In this thesis we show that cross-lingual evidence on association strength (in terms of universally valid surface features) can be used as a single resource for different compound analysis methods yielding a solid performance.

## Compound Identification

In order to identify compounds and distinguishing them from phrases or atomic words, we exploit the spelling form of cross-lingual equivalents. The higher the amount of closed compounds among the equivalents, the higher the degree of compoundhood.

## Compound Parsing

In the spirit of the First Law of Behaghel (1909), we approximate semantic association of target constituents using the word distance of their constituent equivalents in an aligned sentence. The target constituents whose equivalents are further apart, have lower semantic association. We show how nominal compounds of any compound size (in terms of atomic constituents) can be parsed using word distance of constituent equivalents, leading to a binary parse tree.

### 1.4.3. Utilization of Language-independent Morphological Regularities

As third main contribution, we illustrate the utilization of language-independent morphological regularities for sublexical compound analysis, e.g., compound splitting. Previously developed compound splitters are either manual-resource-rich and are thus restricted to a certain target language (e.g., Fritzingler and Fraser (2010)) or rely on cross-lingual supervision (in terms of aligned open compounds) for learning constituent inflection (e.g., Macherey et al. (2011)).

In contrast, we make use of another type of indirect supervision and show that using a linguistic theory about the origin of constituent inflection for closed compounds is a promising starting point towards a language-independent compound splitting method.

In this thesis, we exploit a theory saying that German [linking elements](#) ‘stem from genitive and plural morphemes’ (Neef, 2009). Based on this theory, we develop a [multilingual compound splitter](#) that uses operations for [constituent inflection](#) (e.g., the suffixation of [linking elements](#)) learned from [word inflection](#) (e.g., the suffixation of a genitive marker). We show how this approximation works both for German and related [target languages](#).

#### 1.4.4. Lexical Resources

Apart from the theoretical main contributions described above, this thesis also provides contributions in terms of lexical resources.

##### [Europarl Nominal Compoundhood Ratings](#)

In the course of the linguistic criterion inspection and the evaluation of our [compound identifier](#), two native English-speaking experts in linguistics, in particular in the area of [compoundhood](#), annotated [nominal compounds](#) with ratings about their [compoundhood status](#) and the validity of some [linguistic criteria](#) for [compoundhood](#) in 395 sentences in the [parallel EUROPARL corpus](#)<sup>1</sup>. We will call this [compoundhood](#) gold standard the [Europarl Nominal Compoundhood Ratings \(ENCR\)](#). More details about the [ENCR](#) will follow in Section [10.1.1](#).

##### [Europarl Nominal Compound Database](#)

As result of our [cross-lingual compound identifier](#), we provide a database with English [nominal compounds](#) of any [compound size](#) and their [cross-lingual equivalents](#). As grounding source, we use a part of EUROPARL, comprising 10 European languages, as will be described in Chapter [9](#). We will call this database the [Europarl Nominal Compound Database \(ENCD\)](#). Besides the [word forms](#), the [ENCD](#) contains information about [lemmas](#), [PoS](#), [split points](#), etc. More details about the [ENCD](#) will follow in Chapter [12](#). As will be exemplified for the [cross-lingual compound parsing](#) in Part [E](#), this resource provides useful information for [cross-lingual compound analysis](#). Moreover, it served as basis for theoretical research on monolingual and [cross-lingual compounding](#). The resource was also used in a linguistic study on deverbal [compounds](#) in *English* and *Romanian* (Iordachioaia, 2017).

---

<sup>1</sup>[statmt.org/europarl](http://statmt.org/europarl)



## Compound Parsing Gold Standards

For evaluating the various [cross-lingual compound parsing](#) methods, being applied to EUROPARL, we created several gold standards of [parsed nominal compounds](#) in EUROPARL. These datasets are publicly available and will serve for training and evaluating both monolingual and [cross-lingual compound parsers](#).

## 1.5. Outline of the Thesis

This thesis is structured as follows. There are seven parts: the preface (Part [A](#)), a theoretical description about the nature of compounds (Part [B](#)), [compound identification](#) (Part [C](#)), [compound splitting](#) (Part [D](#)), [compound parsing](#) (Part [E](#)), the overall bottom line of the thesis (Part [F](#)) and finally the appendix (Part [G](#)).

**Part A - Preface:** In the preface of this thesis, we introduce the topic, describe the thematic structure ([1.1](#)), the motivation ([1.2](#)), the overview of research questions ([1.3](#)) and the main contributions ([1.4](#)). Finally, this section gives an outline of what will follow in the subsequent parts and chapters ([1.5](#)).

The two thematic areas (outlined in Section [1.1](#)), the **determination of compoundhood** and the **structural analysis** comprise several parts.

### Determination of Compoundhood

**Part B - Nature of Compounds:** We will describe the theoretical view on the nature of compounds.

In [Chapter 3](#), we will talk about some common characteristics and general aspects of [compounds](#).

The controversy about the definition of [compounds](#) will be discussed in [Chapter 4](#). Besides a small collection of different [compound](#) definitions ([4.1](#)) and some general issues for the definition ([4.2](#)), we will present and discuss different kinds of [linguistic criteria](#) mentioned in linguistics literature.

In [Chapter 5](#), some [cross-lingual](#) observations on [compounding](#) will be presented, e.g., the phenomenon of [parallel compounding](#) ([5.1](#)) or phrasal translations ([5.2](#)).

**Part C - Cross-lingual Compound Identification:** In this part, we will address the task of [compound identification](#) and exploit some [cross-lingual](#) observations.

After **introducing** the **identification** part (**Chapter 7**), we will briefly outline **previous and related work** on **compound identification** in **Chapter 8**, and we will present the **parallel corpus** that will be the basis for all subsequent experiments (**Chapter 9**).

In **Chapter 10** we perform some **pilot studies** concerning **linguistic criteria** for **compoundhood** (10.1) and **cross-lingual compounding** (10.2).

Then, in **Chapter 11**, the **cross-lingual compound identification** method will be described.

The result of applying the **identifier** to EUROPARL, i.e., the **Europarl Nominal Compound Database (ENCD)**, will be presented in **Chapter 12**.

Some **experiments** for assessing the quality of our **identifier** and the potential of the **cross-lingual** approach will be explained in **Chapter 13**.

Finally, the **identification** part is **concluded** (**Chapter 14**).

## Structural Analysis

**Part D - Multilingual Compound Splitting:** The first subtask of **structural analysis** will be presented in this part.

After **introducing** the **compound splitting** part (**Chapter 15**), we will outline **related work** in **Chapter 16**.

The **multilingual compound splitter** exploits automatically learned morphological operations, represented as **Morphological Operation Pattern (MOP)**, which will be described in **Chapter 17**.

The **main method** for **MOP**-based **compound splitting** will be explained in **Chapter 18**.

The **intended compound splitting** based on **Dsim** will be outlined in **Chapter 19**.

Finally, we will describe the **extrinsic evaluation** method using the **downstream task RTE** in **Chapter 20**, and **conclude** the **compound splitting** part (**Chapter 21**).

**Part E - Cross-lingual Compound Parsing:** The second subtask of **structural analysis** will be presented in this part.

After **introducing** the **compound parsing** part (**Chapter 22**), we will outline **related work** in **Chapter 23**, and we will present a **pilot study** in **Chapter 24**,

in which we will illustrate the potential of **cross-lingual** evidence for **compound parsing** with a system that relies on **Aligned Phrase Patterns (APPs)**.

The **main methods** for **cross-lingual compound parsing** will be described in **Chapter 25**. These methods are based on the **AWD** metric (25.1). Besides a **deterministic bottom-up parsing (DBUP)** method (25.2), there will be two non-deterministic approaches that accumulate plausible **parse trees** across languages, the **non-deterministic full tree accumulation parsing (NFTAP)** (25.3.2) and the **Non-deterministic Subtree Accumulation Parsing (NSTAP)** (25.3.3).

Finally the **compound parsing** part will be **concluded** in **Chapter 26**.

**We end this thesis with a final concluding part (Part F) and provide an appendix (Part G).**

**Part F - Bottom Line of the Thesis:** In this part, we will **summarize** the thesis (27.1), **conclude** our research (27.2), **try to answer** the research questions posed in Section 1.3, and finally discuss some possible ways for **future work** (27.3).

**Part G - Appendix:** In this part, we will provide additional information that is useful for understanding the content of the thesis but not necessary.

In **Appendix A**, we will describe the characteristics and the compilation of **universal surface patterns (USPs)**, language-independent and generalized **PoS Patterns** that will be used as format for **APPs** in **compound identification** (e.g., Section 10.1) and **compound parsing** (i.e., Chapter 24).

In **Appendix B**, we will present the morphological operations for **German constituent inflection** collected by Langer (1998), that have been used in previous work on **compound splitting**.

There are different representation forms for the result of **compound splitting**, e.g., as list of **constituent forms** (i.e., the **split point format (SPF)**). In some cases, the compilation of the **SPF** is non-trivial. Several ways for **compiling SPFs** will be discussed in **Appendix C**.

In **Appendix D**, **alternative gold standards on compound splitting** that have not been considered in our experiments will be outlined.

Finally, in **Appendix E**, we will show all annotation guidelines for the **Europarl Nominal Compoundhood Ratings (ENCR)** gold standard, presented in Section 10.1.1.

*1. Introduction to the Thesis*

Part B.

Nature of Compounds



## 2. Introduction to Nature of Compounds

In this chapter, we define and discuss the characteristics and the nature of the main subject of this thesis: the **compound**. Although the definition of **compounds** is discussed controversially in linguistics literature (as will be discussed in Chapter 4), the following subsection (2.1) should provide a basic description, which serves for understanding the subsequent chapters of this part.

The main intention of this part is to give a brief overview about the complex topic of **compoundhood** and **compounding**, serving as background and motivation for the subsequent parts of this thesis.

We restrict to an outline of the nature of **compounds**, because providing a profound and exhaustive study about **compounds** would exceed the scope of this thesis. For a more detailed discussion on the nature of **compounds**, the following works form a helpful starting point: Bauer (1983, 2003, 2006), Booij (2005), Di Sciullo and Williams (1987), Downing (1977), Levi (1978), Liberman and Sproat (1992), Nakov (2013), Warren (1978), and the Oxford Handbook of Compounding (Lieber and Štekauer, 2009). Moreover, Nakov (2013) recommends the extensive Compound Noun Bibliography<sup>1</sup>.

### 2.1. Basic Description

A **compound** is a **complex lexeme** composed of several **atomic lexemes**, which are called ‘**constituents**’ (Bauer, 2003), for example *notebook*, *football match* or *soup tureen*. The **constituents** can be broken up into **modifiers** (or **non-heads**, in *English* usually the non-final **constituents**) and **head** (in *English* usually the final **constituent**) (3.6). **Compounds** can be written as one **word** (i.e., **closed compound**), e.g., *flowerpot* or *flower-pot* or as a **Multi-Word Expression (MWE)** (i.e., **open compound**), e.g., *flower pot* (3.5). These spelling conventions mainly differ with respect to the language. Germanic languages

---

<sup>1</sup>[http://www.cl.cam.ac.uk/~do242/Resources/compound\\_bibliography.html](http://www.cl.cam.ac.uk/~do242/Resources/compound_bibliography.html)

such as German or Dutch usually construct **closed compounds**, whereas English creates **open compounds** (3.9).

## 2.2. Outline

In **Chapter 3**, we will discuss some **general aspects of compounds**. **Compounds** (at least **open compounds**) are a type of **Multi-Word Expressions (MWEs)**. In Section 3.1, we describe alternative types of **MWEs**. The productivity of **compounds**, as described in various corpus studies, is outlined in Section 3.3. The possible functions of **compounding** are discussed in Section 3.4. In previous literature, there are different ways of naming different types of **compounds**. The naming conventions adopted in this thesis are described in Section 3.2. The various spelling forms for **compounds** (i.e., **open**, **hyphenated**, **closed** and finally mixed spelling forms) will be explained in Section 3.5. The different **constituent types** will be presented in Section 3.6, e.g., the **head** of a **compound** (3.6.1) and the different types of headedness (3.6.2). A complexity of **compounds** that arises with three or more **constituents** is structural ambiguity (3.6.4). **Compounds** can be grouped into different classes. In Section 3.7, we will present different classes of **compounds** (including some minor groups such as neoclassical **compounds** (3.7.2), **phrasal compounds** (3.7.3)) and a universal taxonomy (3.7.1). Although this thesis does not address the semantic analysis of **compounds**, a brief overview of the semantics of **compounds** is given in Section 3.8: the compositionality of **compounds** (3.8.1), the implicit **semantic relation** that holds between **modifier** and **head** (3.8.2) and finally a phenomenon of semantic equivalence between different syntactic structures, the **semantic indeterminacy** (3.8.3). Finally, in Section 3.9, we will have a look at **compounding** in different **compounding** languages that are relevant in the remainder of this thesis: English (3.9.1), German (3.9.2), Dutch (3.9.3) and Afrikaans (3.9.4).

The **definition of compounds** is discussed highly controversially in previous literature. In **Chapter 4**, we will outline this discussion based on Lieber and Štekauer (2009) and Nakov (2013). We will cite various definitions of **compounds** (4.1) and will describe the two key issues for the definition problem (4.2). In the subsequent sections, different types of **linguistic criteria** will be presented: orthographical (4.3), morphological (4.4), phonetic/prosodic (4.5), syntactic (4.6) and semantic (4.7) criteria.

An elementary aspect of **compound analysis** addressed in this thesis is **cross-linguality**, i.e., how do **cross-lingual equivalents** of English **target compounds** look like and how can this information be used for the **analysis of compounds**. Therefore, we will consider **com-**



**pounding** from a **cross-lingual perspective** in **Chapter 5**, i.e., we will describe some **cross-lingual** observations that we made when investigating translations of **compounds** in different languages in a **parallel corpus** (Chapter 9). We observed that **compounds** are translated to **compounds** (5.1) or to paraphrases (5.2). Another type of **compound** translations are asymmetric (i.e., non-literal) translations (5.3): aspect alternations (5.3.1), **atomic** translations (5.3.2) or **constituent swapping** (5.3.3). Finally, we will discuss **cross-lingual** indicators for various **compound analysis** tasks, including **compound identification** (5.4.1), **compound splitting** (5.4.2), **compound parsing** (5.4.3), the prediction of **semantic indeterminacy** (5.4.4) or the prediction of the implicit **semantic relations** (5.4.5).

Finally, **Chapter 6** summarizes and concludes this part.

## *2. Introduction to Nature of Compounds*

## 3. General Aspects

Compounding is a phenomenon that is studied extensively in linguistic literature. Also in computational linguistics, **compounds** are enjoying more and more attention (Hendrickx et al., 2013, Ó Séaghdha, 2008).

### 3.1. Multi-Word Expressions

**Compounds** (at least **open compounds**) are a type of **MWEs**, i.e., fixed expressions composed of several **words**. Other types of **MWEs** that are **not** considered as **compounds** include:

**Verb-particle constructions (VPCs):** These are combinations of a base verb and a particle or preposition. These elements can be placed contiguously (as in *put off*) or discontinuously with several intervening **words** (as in *turn the light off*) (Vincze et al., 2011). In particular in German, VPCs can be separated by a lot of content (i.e., the base verbs is placed very early in the sentence, whereas the particle is usually placed in the sentence-final position); for example *Er malte das Gemälde den ganzen Morgen ab* ‘He depicted the painting all morning’.

**Idioms:** The meaning of idioms is not based on the meaning of their parts (Nunberg et al., 1994, Sag et al., 2002). While they mostly have a regular syntax and morphology, the semantics is unpredictable (e.g., *to kick the bucket* meaning *to die*) (Vincze et al., 2011).

**Proverbs:** They express some important wisdoms, e.g., *The early bird catches the worm* (Vincze et al., 2011).

**Light|Support verb constructions:** These are combinations of a nominal and a verbal element. The noun is interpreted literally, whereas the verb has lost its literal sense to some extent, e.g., *to give a lecture*, *to come into bloom* (Vincze et al., 2011).

**Named entities:** These expressions are usually capitalized in English and refer to a unique entity (i.e., proper names). There are different categories of references, usually **person** (e.g., prename, surname, nickname, titles), **organization** (e.g., companies, government, organisations, committees, etc), **location** (e.g., cities, countries, rivers, etc) **date** and **time** expressions (Mansouri et al., 2008).

## 3.2. Naming Convention

### 3.2.1. Inconsistent Naming in Previous Literature

There are various expressions for denoting **compounds** and subtypes of them (e.g., those having a nominal **head**).

Lauer (1995b) collected the following terms that can be roughly interpreted as **complex lexemes** having a nominal **head**:

- *compound nominal*
- *nominal compound*
- *compound noun*
- *complex nominal*
- *nominalization*
- *noun sequence*
- *compound*
- *noun compound*
- *noun-noun compound*
- *noun+noun compound*
- *noun premodifier*

Nakov (2013) defined long sequences of nouns that act as a single noun as *noun compounds*, whereas the same sequence is called *nominal compound* by Downing (1977).

In contrast, Schulte im Walde et al. (2012) defines German **noun compounds** as constructions, “where the **head** (as the rightmost **constituent**) is a noun, and the **modifier** can be from a set of various parts-of-speech”.

### 3.2.2. Naming in this Thesis

We decided on the following naming convention, which allows for being most consistent with respect to previous literature about different types of **compounds**. This naming convention is also in line with Baldwin and Kim (2010), Nagy T. et al. (2011) and Constant et al. (2017).

**Homogenous compositions**<sup>1</sup> of a **word** category  $\Theta$  is called ‘ $\Theta$  *compound*’, i.e., a composition of nouns is a **noun compound**, a composition of verbs is a **verb compound** and a composition of adjectives is an **adjective compound**. If the **compound size** (in terms of **atomic constituents**) is known, we can specify the **size** and call it a ***k*-noun Compound** (kNC), for example a **three-Noun Compound** (3NC).

For a **head-driven** naming of **compounds** where the **modifier** is underspecified, we use a relational adjective. For example, a **compound** with a nominal **head** is called ‘**nominal compound**’, with a verbal **head** a ‘**verbal compound**’ and with an adjectival **head** an ‘**adjectival compound**’. Thus, a **noun compound** is a **nominal compound** with nominal **modifiers**.

In the case that the **heterogenous PoS** of all **constituents** are specified, we can list all of them as **hyphenated modifiers**. For example, a **nominal compound** that has an adjectival **modifier** is called an **adjective-noun compound**.

We refer to a **compound** that has two **constituents** of any type as **binary compound** (BC), to a **compound** having three **constituents** as **ternary compound** (TC) and to a **compound** with *k* **constituents** as ***k*-ary compound** (kC).

A final type of nominal **MWEs** are the **complex nominals** including a preposition or other **functional** markers between the nominal **constituents**. Complex nominals are often found in Romance languages, e.g., the Italian *succo di limone* ‘lemon juice’ or *porta a vetri* ‘glass door’ (Baldwin and Kim, 2010). Similar constructions occurring in English (e.g., *part of speech* or *hall of fame*) are also considered as **complex nominals**. While this expression type will be discussed in several parts of this thesis, we do not consider **complex nominals** as **compounds** but as phrasal constructions.

### 3.3. Productivity and Corpus Distribution

In many languages, **compounding** is (one of) the most productive **word** formation types. Even 2-year-olds can understand and 4-year-olds can produce new **words** by using **compounds** consisting of two morphemes (Clark, 1981). As a consequence, **compounds** are a very common **word** type but many occur with a very low **token** count, which has been shown in various corpus studies.

In a study about English neologisms between 1941 and 1991, Algeo and Algeo (1993) observed that 68% of newly created **lexemes** are **compounds**. While the frequency of **compound types** is high, the frequency of individual **compounds** (i.e., their relative

---

<sup>1</sup>Compositions include both **closed compounds** and **open** sequences.

token frequency) is low. In another English corpus study, Baldwin and Tanaka (2004) observed that only 2-4% of the corpus tokens form constituents of Noun Compounds (NCs) (which is in line with observations made by Ó Séaghdha (2007)), e.g., 2.6% in the *British National Corpus* (while they counted 256K NC types given 939K word types), 3.9% in the *Reuters corpus* or 2.9% in the *Mainichi Shimbun corpus* (Nakov, 2013). Moreover, Ó Séaghdha (2008) observed that the occurrence of NC types follows the Zipfian distribution; and more than 50% of the two-Noun Compounds (2NCs), e.g., *car park*, in the BNC are hapax legomena (Kim and Baldwin, 2006).

A similar pattern of productivity and corpus frequency holds for other languages. In an analysis of the German APA corpus, Baroni et al. (2002) found that almost half (47%) of the word types were compounds. At the same time, the compounds accounted for a small portion of the overall token count (7%), which suggests that many of them are rare (83% of the compounds had a corpus frequency of 5 or lower).

Being abundant as a phenomenon but scarce in terms of individual examples (i.e., the combination of high type frequency and low token frequency) makes the analysis of NCs particularly problematic for statistical techniques that need high token frequencies to make accurate predictions. Data sparsity is expected to lead to low performance. As a consequence, compositional approaches to automatic processing are indispensable, because listing all possible compounds in a dictionary would be as infeasible as listing all possible adjective-noun combinations. Even frequent NCs that have a BNC frequency of 10 and more are covered by only 27% using static English dictionaries (Tanaka and Baldwin, 2003).

## 3.4. Functions of Compounds

NCs have an even higher corpus frequency in technical and scientific domains (e.g., as medical terms or in **patent documents**), because they are often used in complex domain-specific terminology, as well as in titles and abstracts, because they can be used for expressing long and complex phrases with a concise lexeme (Nakov, 2013). “Novel compounds are used as a text compression device i.e., to pack meaning into a minimal amount of linguistic structure, as a deictic device, or as a means to classify an entity which has no specific name” (Lapata and Lascarides, 2003). For example, the complex NP ‘*area for parking the car while attending a football game*’ can be transformed to the two-Noun Compound (2NC) *football parking* (Wisniewski, 1997).

## 3.5. Spelling

### 3.5.1. Closed Compounds

#### Language Distribution

**Closed compounds** are the most frequent spelling form of **compounds** in various **closed compounding languages**, which are spread across several language families around the world, such as **Germanic languages** (e.g., German, Dutch, Swedish, Afrikaans, Danish, Norwegian, Frisian, ...), **Uralic languages** (e.g., Estonian, Finnish, ...), **Hellenic languages** (e.g., Modern Greek), **Slavic languages** (e.g., Czech, Russian, Slovak, ...) and many more.

In **open compounding languages** such as English, **closed compounds** can also occur as an accepted spelling form. However, in most cases, English **closed compounds** are less frequent and mostly lexicalized such as *textbook*, *newspaper* or *Sunday* (Nakov, 2013) and neologicistic **compounding** is usually realized by **open compounds**.

#### Constituent Inflection

A morphological feature that is relevant in particular for **closed compounds** is **constituent inflection**, which one of the **constituents** (usually the non-final **modifier**) undergoes. This includes various morphological operations such as **word-final truncation** to the **word stem** (e.g., in German *schreiben + Heft* → *Schreib|heft* ‘to write + book → writing book’), **word-internal vowel adaption** (e.g., the German Umlautung as in *Mutter + Rente* → *Mütter|rente* ‘mother + pension → mother’s pension’) or **word-final suffixation** (i.e., adding so-called **linking elements** as in the German *Kind + Lied* → *Kinder|lied* ‘child + song → children’s song’).

For **open compounds**, **constituent inflection** is much less frequent as in English (e.g., *girls club*), which will be discussed in Section 3.9.1.

### 3.5.2. Hyphenated Compounds

From the perspective of automatic processing, **hyphenated compounds** (e.g., *health-care*) can be considered and treated as a trivial form of a **closed compound**. **Identification** of the **compound** in running text is trivial because it is already grouped in one **word**. Moreover, there is no need for decompounding (as done with **closed compounds**, see the **compound splitting** Part D), because the hyphens already indicate the **split points**, i.e.,

the boundaries of the concatenated **constituents**. We refer to hyphens that have this function as **split point markers**.

As discussed by Nakov (2013), **hyphenated compounds** are used for special types of **compounds**, such as **copulative compounds** (e.g., *Bosnia-Herzegovina*), **appositional compounds** (e.g., member-state) or for grouping the **modifiers** in larger **compounds** (e.g., *law-enforcement officer*).

#### 3.5.3. Open Compounds

**Open compounds** are the main spelling form of **open compounding languages** such as English: each **constituent** is written as a single **word**, separated by whitespace (e.g., *chicken soup pot*). As a consequence, the determination of **open compounds** occurring in running text is challenging: what is the start and end point of the **open compound** and how can it be distinguished from phrases? Some possible distinctive criteria will be discussed in Chapter 4.

As discussed by Nakov (2013), there are also some **open compounds** in commonly **closed compounding languages** such as Dutch, mainly due to the influence of English or typographical errors. However, a false spelling can be problematic due to semantic ambiguity, e.g., while the Norwegian *røykfritt* means ‘no smoking’, the phrase *røyk fritt* has even the opposite meaning: ‘smoke freely’.

#### 3.5.4. Mixed Spelling Forms

It should be noted mixed spelling forms are found for **compounds** that have three or more **atomic constituents**, e.g., a combination of **closed** and **open compounds** (as in *database connection*), of **closed** and **hyphenated compounds** (as in *Flughafen-Sperrung* ‘airport closure’) or of **open** and **hyphenated compounds** (as in *second-hand clothes*).

### 3.6. Constituents

The **constituents** of a **compound** are either the **HEAD** (3.6.1) or a **MODIFIER**, which is also called **NON-HEAD** (3.6.3). The order in which the **constituents** are expressed is meaningful, i.e., the position of a **constituent** determines its **constituent type** (**modifier** or **head**). The **head** determines the main category of the **compound**, whereas the **modifier** specializes its meaning. For example, while a *birdcage* is a cage for birds, a *cagebird* is a pet bird living in a cage. The **constituents** can be instances of various **PoS** combinations,



some of which are shown in Table 3.3 for English **compounds**. The most frequent PoS combination is the composition of two nouns, the **2NC** (Nakov, 2013).

#### 3.6.1. Head

All compounds have a **head** as lexical core, which inheres most principle semantics (e.g., semantic class), the word category (e.g., noun) and all morpho-syntactic features (e.g., case, gender or number) (Neef, 2009). The term “**head**” usually refers to the syntactic **head**. As will be discussed in Section 3.7, there are **endocentric** and **exocentric compounds**. The former are **compounds** in which the syntactic **head** equals the semantic **head** (e.g., a *policeman* is a *man*). The latter are **compounds** in which the syntactic head is different from the semantic **head**, which is not explicitly expressed (e.g., *birdbrain*, which is commonly understood as a foolish person and not as the brain of a bird). For **endocentric** coordinate **compounds** (discussed in Section 3.7.1), e.g., *producer-director* (an entity being both producer and director), one might argue for a double-**head** or no **head** at all (Lieber, 2009).

#### 3.6.2. Headedness

##### Right-headed **Compounds** and the Righthand Head Rule

For Germanic languages, the **head** is usually the right-most **constituent**, following the **RightHand Head Rule (RHHR)**, e.g., a *birdcage* is a *cage* for birds. Right-headed **compounds** “take their category from the right-hand **constituent**; semantically they are hyponyms of that constituent” (Lieber, 2009).

##### Left-headed **Compounds**

There are some constructions (which may be considered as **compounds**) which are left-headed, for example *vitamin C* (which is a *vitamin* and not a *C*). This type of left-headed **compounds** can be subsumed to cases with a trailing **modifier** which constitutes an identifier, a number or a code: *Route 66*, *Area 51* or *interferon alpha*, borrowings from the usually left-headed Romance languages such as the French *carte blanche*, and constructions with a classifying **head** preceding a proper name, such as *Mount Whitney*, *planet Earth* or *President Trump*. Moreover, some PoS combinations (as shown in Table 3.3) do not follow the **RightHand Head Rule (RHHR)**, for example NOUN+PREP as in *timeout* or VERB+ADV as in *countdown* (Nakov, 2013).

In **complex nominals** of Romance languages, the **head** commonly precedes the **modifier**, as in the Italian *pena di morte* ‘death penalty’ or the Spanish *estado miembro* ‘member-state’ (lit: ‘state member’) (Nakov, 2013). This does not hold for **lexemes** that are realized as **closed compounds** in many languages, such as *airport* (cf. the French *aéroport*) or *motorcycle* (cf. the Spanish *motocicleta*). A more detailed discussion about **parallel closed compounding** will be presented in Section 5.1.1.

### 3.6.3. Modifier

“The higher-level category appears as the **head** of the **compound**, while the **modifier** refers to a feature of the subordinate category that distinguishes the **compound** from other subordinate categories. For example, an *apple tree* is a *tree* that produces apples and not plums, cherries, lemons, etc” (Krott and Nicoladis, 2005).

### 3.6.4. Complex **Compounds** and their Structure

**Compounds** can have more than two **constituents**. Besides **phrasal compounds** (3.7.3), such as *do-it-yourself strategy*, this complexity originates in a recursive construction of a **compound**, i.e., a **constituent** can in turn be a **compound** itself. Theoretically, this recursion can be endless, as illustrated by Nakov (2013) for the **compound** *orange juice*, shown in Table 3.1. However, in practice, **compounds** are usually binary or at most ternary, or are broken down to paraphrases including **compounds** having a lower arity.

orange	juice					
orange	juice	company				
orange	juice	company	homepage			
orange	juice	company	homepage	logo		
orange	juice	company	homepage	logo	update	
orange	juice	company	homepage	logo	update	...

Table 3.1.: Recursive compound construction for *orange juice*

While the **compounds** shown in Table 3.1 are represented in one line, the recursive construction allows for a hierarchical structure, as shown in Figure 3.1.

In analogy to the syntactic ambiguity of sentences (e.g., **PP-attachment ambiguity**), the tree structure of **compounds** having three or more **constituents** is also ambiguous. For example, the **three-Noun Compound (3NC)** *plastic water bottle* can have a **LEFT-** and a **RIGHT-branched** structure as shown in Figure 3.2.

### 3. General Aspects

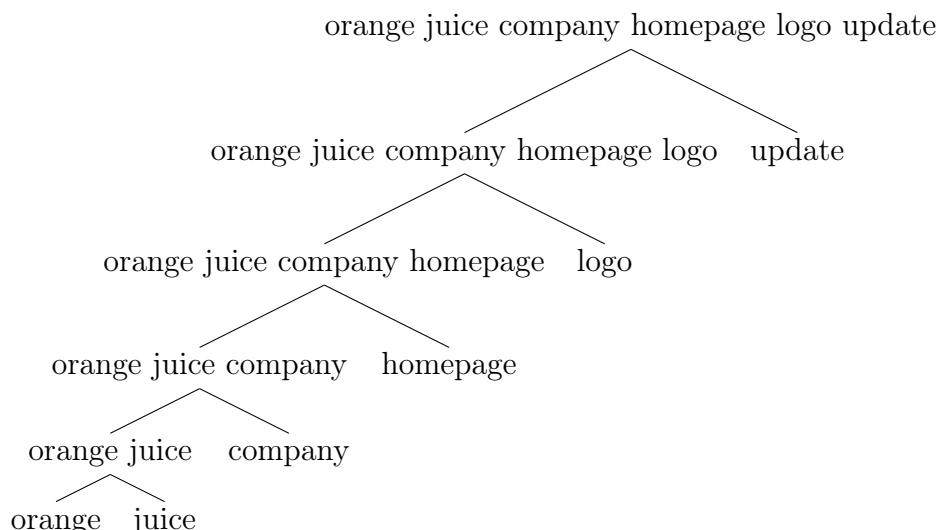


Figure 3.1.: Example of a compound tree structure

With respect to the intended meaning of *plastic water bottle* (i.e., it's a water bottle made out of plastic, instead of a bottle for plastic water), the correct [bracketing](#) is RIGHT-branched (Nakov, 2013).

While the parsing of [3NCs](#) can be considered as a binary classification (i.e., LEFT or RIGHT), there are many more possible structures when the [compound size](#) increases. Table [3.2](#) shows the number of possible binary trees for [compounds](#) having up to 15 [constituents](#).



Figure 3.2.: Tree structure for plastic water bottle

The number of possible binary trees increases with the Catalan numbers (Church and Patil, 1982): [compounds](#) with  $k$  [constituents](#) can be represented by  $Cat_{k-1}$  possible binary trees, where  $Cat_n$  is the  $n$ -th Catalan number as given in Formula [3.1](#).

$$Cat_n = \frac{(2n)!}{(n+1)! \cdot n!} \quad (3.1)$$

Compound size $k$	Binary trees
2	1
3	2
4	5
5	14
6	42
7	132
8	429
9	1430
10	4862
11	16,796
12	58,786
13	208,012
14	742,900
15	2,674,440
$k$	$Cat_{k-1}$

Table 3.2.: Number of possible binary trees for **compounds** with  $k$  **constituents**

As final remark, it should be noted that for some **compounds** there are several syntactic structures that are compatible with the intended meaning. This phenomenon is called **semantic indeterminacy** and will be discussed in Section 3.8.3.

In Part E, we will present several **cross-lingual parsing** methods for determining the correct syntactic structure, that is suitable for a **compound**'s intended meaning.

## 3.7. Compound Classes

### 3.7.1. Universal Taxonomy

Bisetto and Scalise (2005) propose a universally valid taxonomy for classifying different classes of **compounds**, which is briefly outlined in this section. There are many approaches of classifying classes of **compounds** in earlier work such as Bally (1944), Bauer (2001), Bloomfield (1933), Booij (2005), Haspelmath (2002), Marchand (1960), Olsen (2001), Spencer (1991). The advantages and disadvantages of these taxonomies are discussed by Bisetto and Scalise (2005). In the scope of this thesis, we will not go into this discussion but restrict to the universally valid taxonomy proposed by Bisetto and Scalise (2005), “as this seems to be the best thought-out and most cross-linguistically applicable classification available” (Lieber, 2009).

The key criterion for the compound classification is the grammatical relation that

### 3. General Aspects

holds between **modifier** and **head**. These relations are *subordination*, *attribution* and *coordination*; and become the first level of **compound** classes. For each class, there is an **endocentric** (where the syntactic **head** equals the semantic **head**) and an **exocentric** (where the syntactic **head** does not correspond to the (implicit) semantic **head**) version.

**Subordinate compounds** have a complement relation between **modifier** and **head**, e.g., *taxi-driver* (the driver **of** a taxi) or *apron string* (the string **in** an apron). These **compounds** can be both **endocentric** (e.g., *dishwasher*) or **exocentric** (e.g., *cut-throat*). This is quite a productive class in *English* (Lieber, 2009). A very productive subgroup are **compounds** having a (de)verbal **head** (i.e., synthetic **compounds**) such as *truck driver*, *fresh-baked* or *well-preserved*. Another group of **compounds** have a (de)verbal **modifier**: *kick ball*, *call girl*, *attack dog* or *skate park* (Lieber, 2009). English **verbal compounds** also fall in this class: *to air-condition*, *to baby-sit* or *to color-code*. Besides (de)verbal **constituents**, there are also cases of English subordinate **compounds** composed of two non-derived nouns such as *cookbook author* or *gas price*. While the above-mentioned examples are **endocentric**, there are a few English subordinate **exocentric compounds**: *pickpocket*, *cut purse* or *spoil sport*. Although more common in Romance languages such as French (e.g., *porte-parole* ‘spokesperson’ (lit: ‘carry-speech’)), subordinate **exocentric compounds** are not productive in English (Lieber, 2009, Marchand, 1969).

**Attributive compounds** have a relation such that the **modifier** describes an attribute of the **head**. This can be either an adjective (e.g., *blue cheese*) or by a noun that is used metaphorically (e.g., *snail mail* - slowly delivered mail). As discussed by Lieber (2009), attributive **compounds** constitute perhaps the most productive class in English, because the majority of **nominal compounds** that have a nominal **modifier** (i.e., **noun compounds**) are attributive **compounds**, for example *satellite nation*, *sister node* or *key word*. Attributive **adjective-noun compounds** include *barefoot*, *heavy weight* or *long term*. Examples for attributive **adjectival compounds** are *dog tired*, *life long* or *funny peculiar* (Lieber, 2009). As argued by Booij (1992), the process of forming **exocentric** attributive **compounds** (e.g., *bird brain* or *red head*) should be considered as a process of “metonymy at work in languages”. For sure, there are **exocentric** attributive **compounds** being plausibly ambiguous with respect to a literal (i.e., **endocentric**) and non-literal (i.e., **exocentric**) reading, e.g., *birdbrain* - a foolish person vs. the organ of a bird (Lieber, 2009). Attributive **adjective compounds** having a participle **head** of a body part (e.g., *blue-eyed*, *long-*

### 3. General Aspects

*legged, grey-bearded*) have been discussed controversially in literature, because its **heads** cannot occur isolated (e.g., ✗*the man is eyed* vs. ✓*the man is blue-eyed*). While being considered as a suffixed **exocentric compound** (i.e., [*grey-beard*]+*ed*) by Marchand (1969), Hudson (1975) and Ljung (1976) argued that the **head** is still possible but uncommon because of missing informativity. Actually, informative constructions are fine (e.g., ✓*a bearded man*), and therefore, such attributive **compounds** can be considered as being **endocentric** (Lieber, 2009).

**Coordinate compounds** can be considered as a conjunction of their **constituents**, e.g., *poet painter* refers to an entity being both *poet* **and** *painter*, or *singer songwriter* refers to an entity being both *singer* **and** *songwriter*. As discussed by Lieber (2009), coordinate **endocentric compounds** are not common in English. Examples of this class include *spiderman*, *comedy drama* or *king emperor* for nouns, *blue green* and *deaf mute* for adjectives and *trickle irrigate* or *slam dunk* for verbs. A more productive class is coordinate **exocentric compounds**. As discussed by Lieber (2009), in this class, the **constituents** are kind of co-hyponyms (e.g., humans or grammatical relations). Examples of this class include *doctor patient* (*discussion*), *subject verb* (*agreement*) or *father daughter* (*dance*).

Figure 3.3 shows the six-class taxonomy of Bisetto and Scalise (2005) with various examples for each class.

#### 3.7.2. Neoclassical Compounds

Neoclassical **compounds** are constructions where at least one **constituent** is derived from Greek or Latin (which is called a “semi-word” (Scalise, 1984)), e.g., *anthropology* composed of *anthropo* ‘human’ and *logy* ‘science’, i.e., the science of the humans. Bisetto and Scalise (2005) categorized neoclassical **compounds** as subordinate **compounds**, e.g., *hydrophobia* (*hydro* ‘water’ + *phobia* ‘fear’) meaning the fear **of** water. While in particular in the technical or medical domain, new neoclassical **compounds** can be easily formed, Bauer (1998a) raises concerns about considering neoclassical **compounds** productive: is it really possible to produce new neoclassical **compounds** unconsciously and on the fly (Lieber, 2009)? Neoclassical **compounds** (German: *Konfixkomposita*) are also a phenomenon in the German language, e.g., *Thermo|stat* ‘thermostat’.

### 3. General Aspects

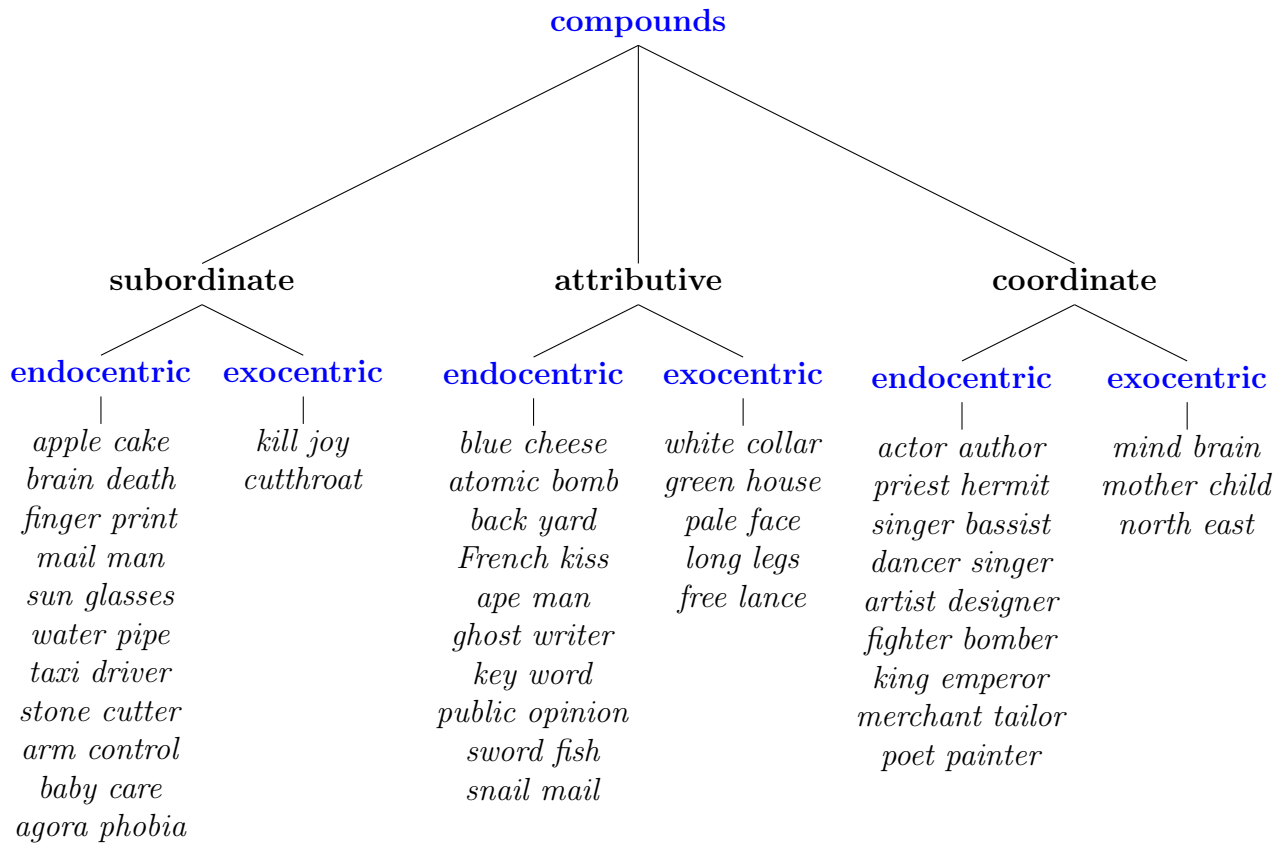


Figure 3.3.: Compound taxonomy by Bisetto and Scalise (2005)

#### 3.7.3. Phrasal Compounds

**Phrasal compounds** can be often found in Germanic languages. These constructions have a phrase as **modifier**, for example *[God is dead] theology*, as discussed in Lieber (1992). Bisetto and Scalise (2005) exemplify **phrasal compounds** with the **compounds** *[floor of a birdcage] taste*, *[punch in the stomach] effect* and *[pipe and slipper] husband*. According to these examples (where the **modifier** is usually understood metaphorically), they argue to consider **phrasal compounds** as attributive **compounds**, i.e., the property of a taste, an effect or a husband is described. However, there are also cases of subordinate **phrasal compounds**, as suggested by (Lieber, 2009), e.g., *over-the-fence gossip* or *in-your-own home care*.

### 3.7.4. Other Classes of Compounds

Nakov (2013) distinguishes two subclasses of coordinate compounds (cf. Section 3.7.1): copulative compounds and appositional compounds.

#### Copulative Compounds

Copulative compounds (known as *dvandva* in Sanskrit) denotes an entity that is the “sum” of the constituents, and distinct from each constituent in isolation, e.g., *Austria-Hungary*<sup>2</sup> (which was a “constitutional union of the Austrian Empire [...] and the Kingdom of Hungary [...] that existed from 1867 to 1918”) or *gerund-participle*.

#### Appositional Compounds

Appositional compounds are characterized by constituents which contribute different aspects of the entity denoted by the compound, e.g., *coach-player* or *sofa-bed*. A *coach-player* is someone being both *coach* and *player* (Nakov, 2013).

#### Reduplication

A minor class of compounds being less productive is the reduplication as in *bye-bye*, *chit-chat* or *walkie-talkie* (Nakov, 2013).

Besides these lexicalized compounds, productive reduplication is mostly used in colloquial spoken English (Lieber, 2009). Reduplication can have an intensifying function (e.g., a *friend friend* as opposed to a *girl friend*). More discussion on reduplication can be found in Hohenhaus (1998).

#### Portmanteaux Compounds

A borderline case of compounds (Neef, 2009) are the portmanteaux compounds (also called ‘blends’). A portmanteaux compound is a creative word coining by merging letters of two lexemes (e.g., a prefix of the modifier with a suffix of the head) of the same word class and the same semantic field (Neef, 2009). However, it is still unclear, what exact regularities are used for the blending (e.g., which order of the constituents). For example, *brunch* composed of *breakfast* and *lunch* or *Merkozy* composed of *Merkel* and *Sarkozy* (Nakov, 2013). Portmanteaux compounds also occur in other languages such as German: *jein* ‘both yes and no’ composed of *ja* ‘yes’ and *nein* ‘no’.

---

<sup>2</sup>[en.wikipedia.org/wiki/Austria-Hungary](http://en.wikipedia.org/wiki/Austria-Hungary)



### 3.7.5. Compound Classes in this Thesis

In this thesis, we do not restrict to a certain **compound** class. Since we focus on **nominal compounds**, the **head** has to be a common noun. Thus, the **target compounds** do not include neoclassical **heads** (3.7.2). Moreover, we do not consider portmanteaux **compounds** to be **nominal compounds**.

## 3.8. Compound Semantics

### 3.8.1. Compositionality

A compositional **compound** is transparent with respect to their **constituents**, i.e., each **constituent** contributes to the intended meaning (i.e., the intended meaning) of a **compound**. As a consequence, **compounds** having a metaphorical sense (e.g., *ivory tower*) or whose **constituents**' composition only becomes transparent when having enough etymological or linguistic background are considered as being non-compositional or semantically opaque. For example, *ladyfinger* is usually interpreted as a metaphor and thus non-compositional, i.e., cookies looking like the finger of a lady; or the **compound** *hippopotamus*, derived from the Greek *hippopotamos* (lit: 'river horse'), is also considered opaque.

As discussed by Nakov (2013), compositionality has to be considered as a continuum rather than a clear classification. Levi (1978) argues for five degrees of compositionality:

- Ⓐ **transparent**: *mountain village* or *orange peel*
- Ⓑ **partly opaque**: *grammar school* or *brief case*
- Ⓒ **exocentric**: *birdbrain* or *ladybird*
- Ⓓ **partly idiomatic**: *monkey wrench* or *flea market*
- Ⓔ **completely idiomatic**: *honeymoon* or *duck soup*

Von der Heide and Borgwaldt (2009) defined a compositionality rating scale between 1 (definitely opaque) and 7 (definitely transparent).

### 3.8.2. Semantic Relation

Besides the lexicalized meaning of non-compositional **compounds** and the **word** sense ambiguity of the **immediate constituents**, a **compound** meaning also depends on the implicit **semantic relation** that holds between **modifier** and **head**. Heringer (1984) presents nine possible readings for the German **noun compound** *Fischfrau* (lit: ‘fish woman’), listed below:

- woman that sells fish
- woman that has brought fish
- woman standing close to fish
- woman eating fish
- woman looking like a fish
- spouse of a fish
- woman and fish at the same time (i.e. mermaid)
- woman having Pisces as zodiac (German *Fisch*)
- woman as cold as a fish

However, there are much more (virtually infinitely many) readings possible - “any interpretation that is pragmatically sensible is a possible one” (Haspelmath, 2002, Neef, 2009).

A special class of **compounds** are **synthetic compounds** that have a deverbal **head** with an argument-supporting nominal (ASN) reading, as described by Iordachioaia et al. (2016) and in Section 1.2.1. While these constructions are grammatically treated as **compounds**, the **semantic relation** between **head** and **modifier** (usually a verbal complement or adjunct) is restricted to grammatical relations, e.g., the **modifier** *Appetit* ‘appetite’ in the **synthetic compound** *Appetit|hemmer* ‘appetite suppressant’ is interpreted as the internal argument of the verb *hemmen* ‘inhibit’ (Neef, 2009).

### 3.8.3. Semantic Indeterminacy

#### Structural and Semantic Ambiguity

As discussed in Section 3.6, complex **compounds** that have three or more **atomic constituents** are ambiguous with respect to their syntactic structure (e.g., whether a **3NC** is LEFT- or RIGHT-branched).

In general, the syntactic structure of a **compound** correlates with its meaning, as in *natural language processing*, shown in Figure 3.4. A LEFT-branched *natural language*

### 3. General Aspects

*processing* means the processing of natural languages, whereas a RIGHT-branched analysis of *natural language processing* means the natural processing of any language, for example the cerebral processing of a programming language.



Figure 3.4.: Tree structure for ‘Natural Language Processing’

### Semantic Indeterminacy for PP-Attachment Ambiguity

Hindle and Rooth (1993) discussed the resolution of **Prepositional Phrase (PP)** attachment ambiguity in sentences like in Example 1a, where the **PP** *with the telescope* can be attached to the object **noun phrase (NP)** *the man* (meaning that the man has a telescope) or to the verb *saw* (meaning that the telescope is used as instrument in the event of seeing a man).

- (1) a. *I saw the man **with** the telescope*

However, Hindle and Rooth (1993) observed that there is a difficulty in ambiguity resolution, because in some cases “there seemed to be a systematic semantically based indeterminacy about the attachment” (Hindle and Rooth, 1993). In Example 2a, there is no difference in meaning when attaching the **PP** *in one neighborhood* to the object **NP** *the same bars* or to the verb *to frequent*: *frequenting [the same bars in one neighborhood]* infers that the frequenting event also takes place *in one neighborhood*.

An alternative case is given in Example 2b. Here, the problem is that “signing an agreement usually involves two participants who are also parties to the agreement.” (Hindle and Rooth, 1993).

- (2) a. *... known to frequent the same bars **in** one neighborhood*  
 b. *We have not signed a settlement agreement **with** them.*

As a conclusion, Hindle and Rooth (1993) coined the **term semantic indeterminacy** for **PP** attachment as:

### 3. General Aspects

*an attachment is **semantically indeterminate** if situations that verify the meaning associated with one attachment also make the meaning associated with the other attachment true*

In their experiments, Hindle and Rooth (1993) observed 77 / 880 (= 8.75%) cases of **semantic indeterminacy** of **PP** attachment.

#### Semantic Indeterminacy for Compounds

Switching from the **PP**-attachment ambiguity on the sentence level to the lexical level, viz., structural ambiguity of **compounds**, we can also observe cases of **semantic indeterminacy**, as discussed by Lauer (1995b), who developed a method for **bracketing 3NCs**. For example, the **3NC** *city sewerage systems*: there is no situation, in which a **LEFT**-branched participant is true but a **RIGHT**-branched is false (or vice versa).

Semantically indeterminate **ternary compounds** (**TCs**) can be considered as being “*both LEFT- and RIGHT-branching, i.e. a dependency should exist between all **word pairs***” (Vadas, 2009). For example, we can describe a relation between all **constituents** in the **3NC** *government policy decisions*:  $w_1 \sim w_2 =$  the *policy of the government*;  $w_1 \sim w_3 =$  the *decisions made by the government*;  $w_2 \sim w_3 =$  the *decisions about the policy*.

This way, **semantic indeterminacy** can be indicated by replacing **TCs** by paraphrasing **NPs**, as shown in Example 3. If replacing the **3NC** *precision navigation systems* in Example 3a with the **NP** in the Examples 3b or 3c does not change the meaning of the initial sentence, the **3NC** can be considered as being **semantically indeterminate**.

- (3) a. *Most advanced aircraft have precision navigation systems.*  
b. **LEFT**: *systems for [precision navigation]*  
c. **RIGHT**: *[navigation systems] that are precise*

Some other examples from the dataset of structure-annotated **3NCs** of Lauer (1995b) are given in Example 4.

- (4) a. *college football player*: ‘a player of college football’ vs. ‘a football player attending the college’  
b. *highway transportation systems*: ‘systems for highway transportation’ vs. ‘transportation systems for the highway’  
c. *computer graphics systems*: ‘systems for computer graphics’ vs. ‘graphics systems for computers’

### 3. General Aspects

In total, Lauer (1995b) observed 35 / 279 cases (= 12.5%) of **semantically indeterminate 3NCs** within his dataset.

It should be noted that **semantic indeterminacy** is a phenomenon not restricted to English only. A German example for **semantic indeterminacy** is the 3NC *Kinder|buch|reihe* ‘children’s book series’, which can be paraphrased both RIGHT-wise (i.e., *eine Buchreihe für Kinder* ‘a book series for children’) and LEFT-wise (i.e., *eine Reihe von Kinderbüchern* ‘a series of children’s books’), and according to the definition of Vadas (2009) there is also a relation between  $w_1$  and  $w_3$ , namely *eine Reihe für Kinder* ‘a series for children’.

Therefore, we conclude the following definition of **semantic indeterminacy** in the context of *k*-ary compounds (kCs):

*a kC is semantically indeterminate between two structures  $\alpha$  and  $\beta$  if situations that verify the meaning associated with  $\alpha$  also make the meaning associated with  $\beta$  true*

#### Dealing with Semantic Indeterminacy

Despite the fact that previous work discussed **semantic indeterminacy** of kCs, to the best of our knowledge, no attempt has been made to include this phenomenon in syntactic analysis. Vadas (2009) argues that in some cases the intended structure is unambiguous. For example, for the NP *American President George Bush*, there are five possible structures (cf. Table 3.2), some of which are semantically plausible: *[American President] [George Bush]* and *[American [President [George Bush]]]*. While both readings refer to the same<sup>3</sup> entity, the intention of the speaker is not to stress George Bush’s nationality but his function as US-President. Therefore, Vadas (2009) chooses not to include **semantic indeterminacy** in his NP structure annotation of the Penn Treebank developed by Marcus et al. (1993), who called **semantically indeterminate NPs** *permanent predictable ambiguity* (ppa).

We believe that it is important to include **semantic indeterminacy** in NLP, e.g., an anaphora resolver needs to know a structural equivalence for finding all possible nested antecedents, e.g., for the 3NC *animal welfare standards* the 2NCs *animal welfare* and *welfare standards*.

We will address the resolution of structural ambiguity in kNCs (i.e., *k-noun Compound parsing*) in Part E. There, we will also take into consideration the cases of **semantic indeterminacy**.

---

<sup>3</sup>George Bush is American, a President and the American President (= US-President). A very unlikely reading for an exclusively RIGHT-branched structure is that there is an American George Bush that becomes President of another country.

## 3.9. Compounding across Languages

### 3.9.1. English

This section outlines the characteristics of English **compounds** discussed by Lieber (2009), i.e., the 18th chapter of the **Oxford Handbook of Compounding** edited by Lieber and Štekauer (2009). While there are many (lexicalized) cases of English **closed compounds**, with respect to the productivity of **compounding**, English can be considered as an **open compounding language**.

#### Word Categories of the Head

PoS Combination	Example
<b>NOMINAL COMPOUNDS</b>	
ADJ + NOUN	<i>hot dog</i>
<b>NOUN + NOUN</b>	<b><i>database</i></b>
PREP + NOUN	<i>underwater</i>
VERB + NOUN	<i>cutthroat</i>
<b>VERBAL COMPOUNDS</b>	
ADJ + VERB	<i>to highlight</i>
NOUN + VERB	<i>to fingerprint</i>
PREP + VERB	<i>to withdraw</i>
VERB + VERB	<i>to freeze-dry</i>
<b>ADJECTIVAL COMPOUNDS</b>	
ADJ + ADJ	<i>dark-blue</i>
NOUN + ADJ	<i>bulletproof</i>
PREP + ADJ	<i>over-eager</i>
<b>ADVERBAL COMPOUNDS</b>	
ADJ + ADV	<i>left-most</i>
NOUN + ADV	<i>headfirst</i>
VERB + ADV	<i>countdown</i>
<b>PREPOSITIONAL COMPOUNDS</b>	
ADJ + PREP	<i>forthwith</i>
NOUN + PREP	<i>timeout</i>
PREP + PREP	<i>without</i>
VERB + PREP	<i>cut-off</i>

Table 3.3.: Possible PoS combinations for English **binary compounds**

There are various **PoS** combinations of English **compounds**, some of which are listed

in Table 3.3, where the PoS pairs are grouped according to the head PoS (e.g., **nominal compounds**, **verbal compounds** or **adjectival compounds**).

#### Compound Classes

Traditionally, English **compounds** have been classified as either **synthetic compounds** or **root compounds**.

English **synthetic** (deverbal, verbal nexus) **compounds** are characterized as having a deverbal **head**, as in *truck driver*, *hard-working*, *home made* or *home improvement*.

English root compounds are defined as being “not **synthetic compounds**” (Lieber, 2009). Other than the **term** suggests, root compounds can also be composed of derived **constituents**, as in *driving school*.

As alternative classification, Bisetto and Scalise (2005) distribute these two classes across three other classes based on the feature of grammatical and **semantic relations**, as detailed discussed in Section 3.7.1.

#### Constituent Inflection and Linking Elements

In general, English is a morphologically poor language and thus hardly realizes **word inflection**. In the case of plural marking only the **head** gets inflected (i.e., *dog beds* instead of *dogs bed*). However, there are two suffixes that are very infrequently appended to the **modifier**: an *s*-suffix, as in *parks department*, and a possessive marker *'s*, e.g., *children's hour*. As argued by Marchand (1969), these suffixes can be considered as having no certain function but are used as **linking element** like for other Germanic languages such as German (Section 3.9.2), Dutch (Section 3.9.3) or Afrikaans (Section 3.9.4). Examples that underlines the theory of **linking elements** are *oarsman* (who does not need more than one oar) or *frontiersman* (for which a plural interpretation of *frontier* is not plausible) (Lieber, 2009).

#### 3.9.2. German

This section outlines the characteristics of German **compounds** (*'Komposita'*) discussed by Neef (2009), i.e., the 20th chapter of the **Oxford Handbook of Compounding** edited by Lieber and Štekauer (2009). German is a **closed compounding language**. These **closed compounds** can be very long, such as the famous example

*Donau|dampf|schiff|fahrts|gesellschafts|kapitäns|mütze*

‘Cap of the captain of the Danube steam ship company’.

## Word Categories of the Head

PoS Combination	Example
<b>NOMINAL COMPOUNDS</b>	
ADJ + NOUN	<i>Großbaustelle</i> ‘large construction site’
NOUN + NOUN	<i>Fußball</i> ‘soccer’
PREP + NOUN	<i>Übersee</i> ‘overseas’
VERB + NOUN	<i>Zeigefinger</i> ‘forefinger’
<b>VERBAL COMPOUNDS</b>	
ADJ + VERB	<i>kleinschlagen</i> ‘to knock to pieces’
NOUN + VERB	<i>eislaufen</i> ‘to ice-skate’
PREP + VERB	<i>unterlassen</i> ‘to refrain from’
VERB + VERB	<i>kennenlernen</i> ‘to get to know’
<b>ADJECTIVAL COMPOUNDS</b>	
ADJ + ADJ	<i>bestmöglich</i> ‘best possible’
NOUN + ADJ	<i>hundemüde</i> ‘dog-tired’
PREP + ADJ	<i>unterernährt</i> ‘undernourished’
VERB + ADJ	<i>treffsicher</i> ‘accurate’
<b>ADVERBAL COMPOUNDS</b>	
ADJ + ADV	<i>schlechthin</i> ‘plainly’
<b>PREPOSITIONAL COMPOUNDS</b>	
NOUN + PREP	<i>kopfüber</i> ‘headfirst’

Table 3.4.: Possible PoS combinations for German binary compounds

There are many possible combinations of PoS for German, as shown in Table 3.4, where noun compounds occur most frequently. Verbal compounds with a prepositional modifier (e.g., *unterlassen* ‘to refrain from’) can also be classified as *particle compounds*. “Whether verbal compounds are compounds in a strict sense or something else [...] is generally disputed” (Neef, 2009).

## Compound Classes

An infrequent phenomenon is the *self-compounding* (where head and modifier share the same lexeme) as in *Helfers|helfer* ‘accomplice’ or *Zinseszins* ‘compound interest’ (Günther, 1981). Neef (2009) discusses three non-canonical types of modifiers for German compounds: phrases (i.e., *phrasal compounds* (Meibauer, 2003)) as in *Aber-da-hört-sich-doch-gleich-alles-auf-Blick* ‘the this-puts-a-stop-to-everything look’, acronyms or abbreviations



viations as in *US-Präsident* ‘president of the USA’ or single letters as in *O-Beine* ‘bendy legs’. Neef (2009) concludes that there are no constraints for German **compound modifiers**.

#### Constituent Inflection and Linking Elements

An elementary characteristics of German **compounding** is **constituent inflection**, i.e., the addition of suffixes (i.e., *Fugen|elemente* ‘linking elements’), the Umlautung and the truncation to stems. Barz (2005) observed that 30% of all German **compounds** include **linking elements**.

Using a test with a coordination reduction, Fuhrhop (1998) argues that the **linking elements** are appended to the **modifier** (rather than being an individual **constituent** or prepended to the **head**): *Kapitäns-(mützen) und Admiralsmützen* ‘caps of captains and admirals’.

While German **linking elements** originate from genitive and plural morphemes, nowadays they have almost completely lost their meaning and grammatical functions. This can be illustrated with **modifier lemmas** bearing **linking elements** which do not correspond to their inflectional markers, i.e., which are **unparadigmatic**, e.g., *Liebes|brief* ‘love letter’.

Literature discussed another function of **linking elements**: *euphony* - **linking elements** improve the pronunciation (Donalies, 2005). Some counter-examples, presented by Neef (2009) are: *Mund|pflege* ‘oral hygiene’ vs. *Hunde|epflage* ‘caring of dogs’ and *Wind|park* ‘park of wind turbines’ vs. *Kinder|park* ‘park for children’. In both examples, the **compound** pairs have the same phonological context but different types of **constituent inflection**.

As a consequence, according to some views, **constituent inflection** has no function at all (Barz, 2005). One observation that underlines this opinion is that only 10% of all German **compounds** have more than one **constituent form**, and in most of these cases, there is a predominant form (which is used for production), while other forms are lexicalized (Augst, 1975).

#### Prosody

As discussed in Section 4.5, the prosody of **compounds** in English is usually **modifier-stressed**. This general rule also holds for German, if the **head** is simplex. If the **head** is a **compound** itself, then the **modifier** of the **head compound** gets the primary stress (Wiese, 1996). Thus, this stress pattern can be used for resolving structural ambiguity of

German **compounds** having three or more **constituents** (e.g., by a TTS system), as shown for the 3NC *Lebens|mittel|punkt* in Figure 3.5. Stress on the first **constituent** points to a LEFT-branched structure (Figure 3.5(a)), meaning ‘marker on groceries’, whereas a stress on the second **constituent** points to a RIGHT-branched structure (Figure 3.5(b)), meaning a ‘center of life’.



Figure 3.5.: Structural ambiguity in German with different primary stress

Counter-examples are mostly based on a contrastive function, e.g., *Nord|bahnhof* ‘North station’ is RIGHT-branched but has a primary stress on the immediate **modifier**, because it is opposed to **compounds** like *Westbahnhof* ‘West station’.

### 3.9.3. Dutch

This section outlines the characteristics of Dutch **compounds** discussed by Don (2009), i.e., the 19th chapter of the **Oxford Handbook of Compounding** edited by Lieber and Štekauer (2009). Similar to German, Dutch is a **closed compounding language**. **Compounding** is the most productive **word** formation type in Dutch and can also be applied recursively as in *weers|voorspellings|deskundigen|congres* ‘weather forecast experts conference’.

#### Word Categories of the Head

Similar to German, the most frequent **word** categories are the **content word** types (i.e., nouns, verbs and adjectives), but also functional **heads** as in *voorover* ‘headfirst’ (lit: ‘for-over’). A list of all possible **PoS** combinations is given in Table 3.5. As in all other discussed languages, the most frequent **PoS** combination is that of two nouns, for example *vlees|soep* ‘meat soup’. **Nominal compounds** with a verbal **modifier** show similarly high productivity (e.g., *speelveld* ‘play field’). In contrast, **adjective-noun** constructions are rather infrequent (e.g., *sneltrein* ‘express train’). Moreover, the adjectives are restricted to Germanic adjectives (e.g., none with a Romance origin) (de Haas and

### 3. General Aspects

PoS Combination	Example
<b>NOMINAL COMPOUNDS</b>	
NOUN + NOUN	<i>vleessoep</i> ‘meat soup’
VERB + NOUN	<i>speelveld</i> ‘play field’
ADJ + NOUN	<i>sneltrein</i> ‘express train’
<b>VERBAL COMPOUNDS</b>	
PREP + VERB	<i>opbellen</i> ‘to phone’
NOUN + VERB	<i>rangschikken</i> ‘to sort’
<b>ADJECTIVAL COMPOUNDS</b>	
NOUN + ADJ	<i>vrouwvriendelijk</i> ‘woman friendly’
VERB + ADJ	<i>fluisterzacht</i> ‘whisper soft’
ADJ + ADJ	<i>donkerblond</i> ‘dark blond’
<b>PREPOSITIONAL COMPOUNDS</b>	
PREP + PREP	<i>voorover</i> ‘headfirst’

Table 3.5.: Possible PoS combinations for Dutch binary compounds

Trommelen, 1993). Much less productive (and also hardly recursively applicable) are **adjectival compounds**, such as *steen|rood* ‘stone red’ or *vrouw|vriendelijk* ‘woman friendly’, both having a nominal **modifier**, or *fluister|zacht* ‘whisper soft’, having a verbal **modifier**. The most productive group among the **adjectival compounds** are **adjective compounds**, having an adjectival **modifier**, as in *donkerblond* ‘dark blond’ or *stom|verbaasd* ‘very surprised’ (Don, 2009). **Verbal compounds** are grouped into separable and inseparable **verbal compounds**. The first group is driven by syntactic processes and separates the verb stem from the **modifier** (e.g., a verb particle). For example, the **verbal compound** *op|bellen* ‘to phone’ is separated in the clause *dat ik Iris op probeer te bellen* ‘that I try to phone Iris’, a construction which is not possible for German. While Don (2009) classified particle verbs as **verbal compounds**, we will not consider these types of **complex lexemes** as **compounds** (as discussed in Section 3.1). In contrast, inseparable **verbal compounds** have a **content word modifier** such as the noun *rang* ‘rank’ in the **verbal compound** *rang|schikken* ‘to sort’, as in the clause *dat Annelot de blokken rangschikt* ‘that Annelot arranges the blocks’. However, the formation of new inseparable **verbal compounds** is not very productive (as opposed to separable particle verbs) (Don, 2009).

## Compound Classes

Dutch **compounds** are usually **endocentric**, i.e., the semantic **head** is explicitly expressed (as the syntactic **head**). There are very few **exocentric compounds** (having an implicit semantic **head**), often referring to persons such as *rood-huid* ‘American Indian’ (lit: ‘red skin’) or *zwart-hemd* ‘fascist’ (lit: ‘black-shirt’). Dutch **compounding** also include **synthetic compounds** such as *weersvoorspelling* ‘wheather forecast’, where the **modifier** *weers* ‘wheather’ can be interpreted as the internal argument of the verb related to the **head** *voorspelling* ‘forecast’. We will not discuss Dutch **synthetic compounds** in the scope of this thesis.

## Constituent Inflection and Linking Elements

As for German and many other **closed compounding languages**, there is sometimes a **linking element** added between **modifier** and **head** (i.e., a **constituent inflection** operation applied to the **modifier**), for example *honde|hok* ‘doghouse’. Usually, the additive suffixes used in Dutch **constituent inflection** are:  $\oplus s$ ,  $\oplus e$ ,  $\oplus en$  and  $\oplus er$ .

As for German, the **linking elements** have a functional origin (e.g., case markers) or are remnants of Dutch **words** that originally ended on schwa (either spelled as *e* or *en*). Due to analogy, these suffixes have been appended to other **modifiers**.

For **compounds** such as *deskundigenn|congres* ‘experts conference’, *steden|raad* ‘cities council’ or *boekenk|kast* ‘book case’, the **linking element** seems to be meaningful, i.e., marking the plural interpretation of the **modifier**. However, there are plenty of counter-examples where a plural marker is missing for an obvious plural interpretation (e.g., *boek|handel* ‘book shop’). While we can conclude that the **linking elements** are no plural markers (i.e., they are mostly semantically empty), there is a correlation in between: if a **modifier** form ends on *-e(n)*, the plural form of its **lemma** is marked with *-en*, and if a **modifier** form ends on *-er*, the plural form of its **lemma** is marked with *eren* (de Haas and Trommelen, 1993, Mattens, 1970).

The selection of correct **linking element** is said to be based on analogy in morphology (Krott, 2001). For an adjectival **modifier**, either the stem is used as **constituent form** (e.g., *frisdrank* ‘fresh drink’) or an *e*-suffix (e.g., *wittebrood* ‘white bread’ or *hogeschool* ‘high school’).

#### Prosody

The prosody of Dutch **compounds** is similar to English and German: the main stress is usually on the **modifier**, whereas a secondary stress is on the **head** (Booij, 1995, Langeweg, 1988, Visch, 1990). Exceptions of this rule (i.e., **compounds** having a **head-stress**) include *stad|huis* ‘city hall’ or *boeren|zoon* ‘farmer’s son’. While one reason for **head-stress** is based on the **modifier lemma** (e.g., the **modifier** *stad* ‘city’ is never stressed), “there are no systematic reasons present that could explain these deviant stress-patterns in bipartite **compounds**” (de Haas and Trommelen, 1993, Don, 2009).

#### Headedness

Dutch **compounds** are right-headed, the **head** agrees with the **compound** in PoS, gender, number and semantic class (Trommelen and Zonneveld, 1986). There are only a few lexicalized left-headed **compounds**, such as the verb *schuddebuik* ‘shake with laughter’ (lit: ‘shake-belly’).

#### 3.9.4. Afrikaans

This section briefly outlines the characteristics of Afrikaans **compounds** described by Huyssteen and Zaanen (2004) and by Verhoeven et al. (2014).

#### Word Categories of the Head

The **constituents** of Afrikaans **compounds** can be of various **word** categories, but notably from nouns, verbs, adjectives, and adverbials (Combrink, 1990). In analogy to all other discussed **closed compounding languages**, the most frequent type are Afrikaans **noun compounds**, such as *hondehok* ‘doghouse’.

#### Compound Classes

There are various classes of Afrikaans **compounds** (Huyssteen and Zaanen, 2004, Verhoeven et al., 2014).

**Primary (root) compounds** are composed of two (possibly **constituent-inflected**) **lexemes** (e.g., *leg|kaart* ‘puzzle’)

**Phrasal compounds** have a phrasal **modifier** (e.g., *help-my-fris-lyk-hemp* ‘gym vest’ (lit: ‘help-me-strong-look-shirt’))

**Neo-classical compounds** are usually composed of two bound morphemes, e.g., *bio|logie* ‘biology’

**Synthetic compounds** “are formed by means of affixation based on **word** groups or syntactic constructions, and is in Afrikaans not necessarily verbally based” (Botha, 1981, Huyssteen and Zaanen, 2004). For example, the **compound** *vyf|week|liks* ‘five weekly’ is composed of the numeral *vyf* ‘five’, the noun *week* ‘week’ and the adjectival derivation suffix *liks* ‘-ly’.

**Compounding compounds** are LEFT-branched constructions with at least three **constituents** (usually an adjective and two nouns), for example *mediese|fonds|bydrae* ‘medical aid contribution’, where *mediese fonds* ‘medical aid’ is a fixed **word** group

There are Afrikaans **compound** classes which can occur **hyphenated**, such as **copulative compounds** (e.g., *skilder-skrywer* ‘painter-writer’), **reduplications** (e.g., *speel-speel* ‘play-play’) or **left-headed compounds** (e.g., *prokureur-generaal* ‘attorney-general’).

### Constituent Inflection, Linking Elements and Hyphenation

Afrikaans **compound heads** undergo **word inflection** (e.g., pluralization as in *kisfabrieke* ‘coffin factories’) or **word derivation** (e.g., *katkosagtig* ‘like cat food’).

**Linking elements** joining **modifier** and **head** in Afrikaans **compounds** (as in *besigheids|besluit* ‘business decision’) are well-known (Neijt et al., 2010). However, most Afrikaans **compounds** occur without any **linking element**. In the corpus study of Huyssteen and Zaanen (2004), only 7270 / 40,051 (= 18.2%) **compounds** show **constituent inflection**. The **linking element** distribution of this corpus study is shown in Table 3.6.

Linking element	Example	Distribution
$\oplus s$	<i>verbindings klank</i> ‘connection sound’	5861 (80.62%)
$\oplus e$	<i>honde hok</i> ‘doghouse’	508 (6.99%)
$\oplus ns$	<i>lewens drang</i> ‘life force’	101 (1.39%)
$\oplus er$	<i>kinder skoene</i> ‘children shoes’	90 (1.24%)
$\oplus ens$	<i>nooi<u>er</u> van</i> ‘maiden name’	52 (0.72%)
$\oplus n$	<i>buiten gewone</i> ‘extraordinary’	2 (0.03%)
hyphen	<i>sterre-energie</i> ‘star power’	656 (9.02%)

Table 3.6.: Distribution of linking elements in Afrikaans **compounding**, by Huyssteen and Zaanen (2004)

### 3. General Aspects

Besides with some **hyphenated compound** classes discussed above, hyphenation is also used in the context of vowel accumulation (e.g., *koei-oë* ‘cow’s eyes’) or for structuring long **compounds** (e.g., *diesel|enjin-wipbak|vrag|motor* ‘diesel engine tipper lorry’) (Huyssteen and Zaanen, 2004).

#### Productivity and Compound Size

The agglutinative Afrikaans is a **closed compounding language**, allowing for productive **closed compound** formation. Similarly to other West-Germanic languages (Dutch, Frisian, German and to a far lesser extent English) (Verhoeven et al., 2014), the **closed compounding** in Afrikaans involves the concatenation of two or more **constituents** (usually free morphemes), e.g., *kat* ‘cat’ + *kos* ‘food’ → *katkos* ‘cat food’.

Compound size	Distribution
2	31,358 (78.30%)
3	7993 (19.96%)
4	663 (1.66%)
5	35 (0.09%)
6	2 (0.005%)

Table 3.7.: Distribution of **compound size** in Afrikaans, by Huyssteen and Zaanen (2004)

“Next to derivation, the process of right-headed, recursive **compounding** is the most productive **word** formation process in Dutch and Afrikaans” (Verhoeven et al., 2014). This productivity and the recursive applicability of Afrikaans **compounding** can lead to complex lexical units, such as *strand|hand|doek|stof|fabriek* ‘beach towel cloth factory’. In theory, the **compound size** (i.e, number of **atomic constituents**) for an Afrikaans **compound** is infinite, but most **compounds** are **binary**, as shown in Table 3.7 for a corpus study by Huyssteen and Zaanen (2004), including 40,051 Afrikaans **compounds**<sup>4</sup>. The largest **compound** Huyssteen and Zaanen (2004) observed was

*radio|telefoon|nood|frekwensie|luister|diens|ontvang|toestel*  
‘radio telephone emergency frequency listening service reception device’.

---

<sup>4</sup>A similar distribution of German, Dutch and Afrikaans **compound size** is presented in Chapter 18, Tables 18.5, 18.7 and 18.9.

### *3. General Aspects*



## 4. The Controversy of the Definition of Compounds

The basic ideas of the following discussion are borrowed from the **Oxford Handbook of Compounding** edited by Lieber and Štekauer (2009, chap. 1) and from Nakov (2013).

The definition of **compounds** (i.e., what properties are necessary and sufficient for a linguistic expression to be considered as a **compound**) is highly controversially discussed in linguistics literature and there are hardly any commonly accepted **criteria**.

Even more, not only the definition of **compounds** is controversial, but even the existence of such a **word** formation type. While Bauer (2003) defines a **compound** as “the formation of a new **lexeme** by adjoining two or more **lexemes**”, Marchand (1967) denies the existence of a **compounding word** formation type besides **expansion** and **derivation**. The key feature for Marchand (1967) is the independence of the rightmost **constituents** (i.e., the **head**). If the **head** is a free morpheme, the underlying **word** formation is classified as **expansion** (e.g., prefixed constructions such as *reheat* and **compounds** such as *steamboat*), and if the rightmost **constituent** is a bound morpheme, it is considered as an instance of **derivation** (e.g., suffixed constructions such as *blindness*).

### 4.1. Various Ways of Compound Definition

The following collection is composed of snippets from the vast amount of different ways how **compounding** (or a (noun) **compound**) is defined in both linguistics and **NLP** literature.

**Marchand (1960)**: “when two or more **words** are combined into a morphological unit, we speak of a **compound**”

**Downing (1977)**: “a sequence of nouns which function as a single noun”.

As discussed by Nakov (2013), the problem with this definition is that there are

#### 4. The Controversy of the Definition of Compounds

**words** that are ambiguous with respect to their category, e.g., adjective vs. noun for the **modifiers** in *adult male rat*, and that nouns and (relational) adjectives can be meaning-preserved exchanged, e.g., *linguistic difficulties* vs. *language difficulties*.

**Levi (1978)** defines three types of **complex nominals**:

- **nominal compounds**: *database, chocolate cake, ...*
- **nominalizations**: *dream analysis, truck-driver, ...*
- **nonpredicate NPs**: *electric shock, musical criticism, ...* (i.e., adjective-noun sequences, where the adjective cannot be used predicatively)

These categories clarify the issue discussed for Downing (1977): *linguistic difficulties* is categorized as nonpredicate NP, whereas *language difficulties* is a **nominal compound**.

**Trask (1993)**: “the process of forming a **word** by combining two or more existing **words**: *newspaper, paper-thin, babysit, video game*”

**Lauer (1994)**: “Compound nouns (CNs) are a commonly occurring construction in language consisting of a sequence of nouns, acting as a noun; *pottery coffee mug*”

**Bauer (2003)**: “the formation of a new **lexeme** by adjoining two or more **lexemes**”

**Vincze et al. (2011)**: “a **compound** is a lexical unit that consists of two or more elements that exist on their own. Orthographically, a **compound** may include spaces (*high school*) or hyphen (*well-known*) or none of them (*headmaster*).”

## 4.2. The Key Issues for the Compound Definition Problem

Lieber and Štekauer (2009, chap. 1) pointed out two key issues for why a “satisfying and universally accepted” definition is problematic.

### 1. What kind of units can be used as constituents during compounding?

Lieber and Štekauer (2009) refer to this issue as the “micro question” of the **compound** definition. Starting with Marchand (1960), saying that “[w]hen two or more **words** are

#### 4. The Controversy of the Definition of Compounds

combined into a morphological unit, we speak of a **compound**”, we have to keep in mind that there are morphologically rich languages (such as Slovak) in which **constituents** may be bound morphemes such as stems or roots, which cannot be considered as independent **words**. For example, the **modifier** in the Slovak **compound** *rýchlovlak* ‘express train’ starts with the stem of the adjective *rýchly* ‘fast’ (as in the phrase *rýchly vlak* ‘fast train’): ‘*rýchl*’ (followed by a **linking element** *o*). The lack of inflection in English makes compositional and phrasal structures collapse with respect to the morphological **word** forms. For example, *blackbird* (**compound**) vs. *black bird* (phrase).

A possible solution for this is to switch from **words** to **lexemes** for the units a **compound** is composed of, as proposed by the **compound** definition of Bauer (2003). The **term** ‘**lexeme**’ seems more suitable<sup>1</sup> for both including free and bound morphemes of lexical units, and simultaneously excluding derivational and inflectional affixes.

On the other hand, this way, we partially break down the **compound** definition to the definition of **lexemes**, which also holds some issues. Lieber and Štekauer (2009) mention some problems of finding a universally valid definition of ‘**lexeme**’. How can bound lexical roots (= **lexemes**) be distinguished from derivational affixes? One possible **criterion** is the amount of semantic content: a **lexeme** has more semantic content than a derivational affix. However, there are languages (e.g., Native American languages) in which so-called “lexical affixes” can have as much semantic content as lexical roots (Mithun, 1999). An alternative **criterion** for the **lexeme** definition is the possibility of occurring isolated (as inflected form). However, this **criterion** allows English particle verbs such as *overfly* or *outrun* to be considered as **compounds**, which is unwanted, because the particles *over* and *out* have a different function than *proof* in *proofread* has, as shown in Example 5.

- (5) a. *The plane overflew the field*  
b. \**The plane flew the field*  
c. *The editor proofreads the article*  
d. *The editor reads the article*

## 2. How can compounds be distinguished from phrases?

Lieber and Štekauer (2009) refer to this issue as the “macro question” of the **compound** definition. According to the definition stated by Bauer (2003), a **compound** is a ‘**new**

---

<sup>1</sup>“The **lexeme** is defined as a set of syntactic and semantic features shared by one or several morpho-syntactic elements. Roughly speaking, it contains the kind of information one expect to find in a standard dictionary entry (Wehrli, 1985)

#### 4. The Controversy of the Definition of Compounds

lexeme’. This holds for lexicalized **compounds** such as *blackboard*, which appears to be different from the phrase *black board*: the former **lexeme** can also be used with other colors (cf. *green blackboard* vs. \**green black board*). But what about deictic **compounds** (Downing, 1977), which are used for referring to objects in the situation of utterance; for example, a *tomato bowl* that just happens to hold tomatoes at the moment of utterance might not be regarded as a single **lexeme**. Moreover, many German **adjective-noun compounds** are semantically equivalent to their phrasal counterparts, e.g., *Optimallösung* ‘optimal solution’ vs. *optimale Lösung* (Schlücker and Hüning, 2010). Can we consider these constructions as **compounds**? At least, they have some properties which are often encountered in **compounds**, such as prosodic stress in English or spelling in German.

Another borderline case are **phrasal compounds** such as the *ate-too-much headache* or a *wouldn’t-you-like-to-know-sneer*, which cannot be considered as **lexemes**, while still being classified as **compounds** in literature.

As a conclusion, Lieber and Štekauer (2009) argue that the only way for getting a suitable **compound** definition is to find solid **criteria**. Although, Lieber and Štekauer (2009) observed that there is almost no reliable and universally accepted **criterion**, they mentioned several plausible tests, which are partially valid for some languages or at least deserve closer attention. Below, we will discuss some of these **linguistic criteria** and show examples, where they work and where they fail.

### 4.3. Orthographical/Spelling Criteria

Donalies (2004) proposes the **criterion** saying that **compounds** are spelled together. This condition (i.e., **closed compounding**) is valid for very many languages as discussed in Section 3.5.1. However, it fails for **open compounding languages** such as English (see Section 3.5.3). Even more, in English “[o]rthography, i.e. spelling convention for **compounds** cannot be taken seriously. . . the orthography of English **compounds** is notoriously inconsistent: some **compounds** are written as single **words** (*postcard*, *football*), in others the **constituents** are **hyphenated** (*sound-wave*, *tennis-ball*), and in still others the **constituent** elements are spaced off, i.e. written as two separate **words** (*blood bank*, *game ball*)” (Szymanek, 1998). There are even English **compounds** which can be observed in all discussed variants: as opaque **closed compound** (*flowerpot*), as **hyphenated compound** (*flower-pot*) and as **open compound** (*flower pot*). “It would be inconsistent to believe that *healthcare* and *health-care* are **noun compounds**, while *health care* is not” (Nakov, 2013).

## 4. The Controversy of the Definition of Compounds

Instead, Nakov (2013) argues to treat **closed compounding** as an **indicator** for **compoundhood** rather than a **criterion**.

Finally, the spelling **criterion** is not applicable to languages that do not mark **word boundaries**, such as Chinese or Japanese; in contrast, for some Germanic languages such as German or Dutch, concatenated **lexemes** reliably correspond to **closed compounds** (Nakov, 2013).

### 4.4. Morphological Criteria

#### 4.4.1. Word Inflection

The **head** undergoes **word inflection**, whereas the **modifier** is uninflected. For example, in **adjective-noun** constructions, the **modifier** is always bare for **compounds**, whereas it undergoes **word inflection** (e.g., marking case, number or gender) for phrases, as in German: *Altpapier* ‘scrap paper’ vs. *altes Papier* ‘old paper’, *Jungfrauen* ‘virgins’ vs. *junge Frauen* ‘young women’ or *bestbezahlter Job* ‘best paid job’ vs. *am besten bezahlter Job*.

However, sometimes **word inflection** is still applied to the **modifier** (Lieber and Štekauer, 2009, Nakov, 2013), e.g., as plural marker as in *overseas investor*, *programs coordinator* or *weapons treaty*. Selkirk (1982) argues that pluralized **modifiers** are used for marking the plurality of the **modifier’s** concept. This is in line with some German **compounds** including **modifiers** for which several **constituent inflection** operations are available. For example *Landes|grenze* ‘country border’ (the border of a country) vs. *Ländergrenze* (the border between two countries). But this plural marker is just an optional means and neither a missing marker indicates singularity nor such a marker always indicates plurality (e.g., a *Kinderbett* ‘child’s bed’ is usually not designed for accommodating more than one child).

Sometimes, the **word inflection** of the **head word** differs from that of the isolated **word**. For example, *sabre tooth*, a pre-historic animal, is pluralized to *sabre tooths* rather than ~~✗~~*sabre teeth* (Nakov, 2013).

#### 4.4.2. Constituent Inflection

Usually, the **modifier** undergoes **constituent inflection**. One type of **constituent inflection** operations is the addition of a **linking element** (also called **linking morpheme**, interfix or intermorph). Although, the **constituent inflection** suffixes conform with **word inflection**

suffixes, they are mostly meaningless attachments to the **modifier** or are considered to occur as isolated **constituents** between **modifier** and **head** (Lieber and Štekauer, 2009). For example, in modern Greek, the **modifier** precedes the **linking element** *o*, which is explained as a historical remnant, which is no longer used in Greek **word inflection**.

### 4.5. Phonetic/prosodic criteria

A general rule for English is that **binary compounds** are stressed on the first **constituent** (i.e., the **modifier**), whereas phrases are stressed on the **head**. Chomsky and Halle (1968) define **compounds** as: “the **words** preceding a noun will form a **compound** with it if they receive the primary stress”. However, there are many exceptions (i.e., English **compounds** stressed on the **head**). There has been an intensive discussion in previous literature about possible reasons for why **compounds** happen to be **modifier**- or **head**-stressed.

Prosody can alter across speakers and dialects (Nakov, 2013). One reason can be the contextual and pragmatic clues (Lieber and Štekauer, 2009), i.e., **compounds** in isolation happen to be stressed differently than in context (Bauer, 1983, Kingdon, 1958, Roach, 1983). Stress can also be used as means for sense disambiguation, such as *toy factory* (meaning a factory producing toys) vs. *toy factory* (meaning a factory which is a toy) (Spencer, 2003).

In another explanation, **noun-noun** constructions are distinguished between **attribute-head** constructions (i.e., those where the **modifier** describes an attribute of the **head**, for example *steel bridge*) and **complement-head** constructions (i.e., where the **modifier** is a complement to the **head**, for example *fruit market*). Giegerich (2004) argues that **attribute-head** constructions, mostly stressed on the **head**, are phrases, whereas **complement-head** constructions, mostly stressed on the **modifier**, are **compounds**. In an experiment, Plag (2006) showed that **attribute-head** constructions also occur **modifier**-stressed without being lexicalized.

A final explanation presented by Lieber and Štekauer (2009) is the semantic principle. For example, Jones (1969) proposes three semantic **criteria** for the **modifier** stress: (1) compositionality: if the **compound** denotes more than just the combination of its **constituents** (as in *blackboard*), (2) importance of the **modifier** (as in *birthday*) and (3) contrastivity: when the **modifier** is contrasted with something (as in *flute player* vs. *piano player*). Ladd (1984) argues that the **head** gets deaccented given a **modifier** subcategorizing the **head**, i.e., the semantics of the **head** is only partially relevant for identifying the semantic category of the whole (Nakov, 2013). Lieber and Štekauer (2009) point out

## 4. The Controversy of the Definition of Compounds

that these **criteria** do not hold for all cases, e.g., why should a **head**-stressed *apple pie* be more compositional than a **modifier**-stressed *apple cake*?

Sampson (1980) argues that a MADE\_OF relation between **modifier** and a **head** denoting a solid artifact leads to a **head** stress (e.g., *iron saucepan*), whereas other types of **heads** lead to a **modifier** stress (e.g., *water droplet*) but also concedes some exceptions (e.g., *rubber band* in American English). Olsen (2000) proposes two further **semantic relations** leading to a **head**-stressed compound: TIME (e.g., *summer night*) and LOCATION (e.g., *hotel kitchen*). Finally, Liberman and Sproat (1992) added the cases of PROPER-NAME MODIFIERS (e.g., *Carlsberg beer*) and LEFT-HEADED COMPOUNDS (e.g., *vitamin D*, *peach melba* or *planet Earth*). Exceptions discussed by Lieber and Štekauer (2009) include *winter coat* and *summer school*.

Plag (2006) showed that the stress of newly created **compounds** are often analogous to existing **compounds** the speaker has in mind.

As conclusion, there is a trend that English **compounds** are stressed on the **modifier**, but there are many exceptions that make this **criterion** less reliable for distinguishing **compounds** from phrases.

## 4.6. Syntactic criteria

### 4.6.1. Inseparability

An English **word** sequence is a **compound** if no element (e.g., an adjectival **modifier**) can be inserted between **modifier** and **head**, i.e., **modifier** and **head** are **inseparable**. While *black bird* (irrespective of the spelling, see Section 4.4) can be understood as **compound**, *black ugly bird* is a phrase. For modifying the **compound** *black bird* with *ugly*, the adjective needs to be preposed: *ugly black bird*. This **criterion** seems fairly reliable when disregarding coordinated **modifiers** as in *wind and water mills* (Lieber and Štekauer, 2009).

### 4.6.2. Inability to Modify the Modifier

In English **compounds**, the **modifier** (i.e., the first element) is not able to be modified, whereas this is possible for syntactic phrases. For example, the phrase *social person* (i.e., any person who is social) can be modified as in *very social person*. This is not possible for the **compound** *social policy* (i.e., a certain type of policy dealing with social aspects).

#### 4. The Controversy of the Definition of Compounds

As a consequence, the **word** sequence *very social policy* enforces a phrasal reading (i.e., any policy that is very social).

In **adjective+noun** constructions, this criterion only holds for qualitative **modifiers** but not for relational adjectives, such as (*\*very*) *mortal disease* (Lieber and Štekauer, 2009).

Exceptional cases mentioned by Bauer (1998b) include *Serious Fraud Office* and *instant noodle soup*.

An example for German is *lange Lebenserwartung* ‘long life expectancy’ (where *lange* ‘long’ modifies *Lebens* ‘life’ rather than *Erwartung* ‘expectancy’). Actually, ‘*lange Lebenserwartung*’ yields 42,100 Google<sup>2</sup> hits, whereas ‘*hohe Lebenserwartung*’ ‘high life expectancy’, which conforms with this syntactic **criterion**, yields 76,100 Google hits.

##### 4.6.3. Inability to Replace the Head with the Pronoun *one*

The **head** of a **compound** cannot be replaced by the pronoun *one*, while this is possible for phrases (Bauer, 1998b). For example, *brown dog* can be transformed to *brown one*, whereas *black bird* (as a **compound**) cannot be paraphrased as *\*black one*. As rare exception, Bauer (1998b) mentions the **compounds** *riding horse* and *carriage one* (where the **head** is an anaphora for *horse*).

## 4.7. Semantic criteria

Nakov (2013) discusses three types of semantic criteria.

### 4.7.1. Permanence

The first **criterion** describes the status of the relationship between **modifier** and **head**. This relationship has to be permanent as in *desert rat* (i.e., a rat always living in the desert). However, this **criterion** does not hold for **compounds** describing short events such as *heart attack*.

### 4.7.2. Non-compositionality

Bauer (2006) argues that a **compound** needs to be at least partially non-compositional, i.e., they have a special (implicit) meaning. For example, the **compounds** *wheel-chair*

---

<sup>2</sup>google.de



#### 4. The Controversy of the Definition of Compounds

and *pushchair* have **modifiers** which denote a property of each other (i.e., a *wheel-chair* can be pushed and a *pushchair* has wheels). However, compositionality is a continuum rather than a clear category: there are more or less compositional **compounds**, ranging from the non-compositional *honeymoon* over the intermediate cases such as *boy friend* to the compositional **compounds** such as *mouse trap*. This makes it hard to use non-compositionality as a **criterion** for **compoundhood** - given that **compoundhood** itself can be considered as a clear category rather than a continuum.

##### 4.7.3. Lexicalization

This **criterion** describes the degree of lexicalization of a lexical unit, i.e., how far can a **word** sequence represent a single lexical entry (Nakov, 2013). The more non-compositional a **compound**, the higher the degree of lexicalization. Moreover, the more lexicalized a term, the higher the chance that it is spelled as one **word** (i.e., as a **closed compound**). For example, the **closed** *bathroom* is considered to be more lexicalized than the **open** *game room* (Nakov, 2013).

#### 4. *The Controversy of the Definition of Compounds*

# 5. Cross-lingual Observations of Compounds

This chapter provides a qualitative study and discussion on the [cross-lingual](#) character of [compounding](#), i.e., how English [compounds](#) can be expressed in other languages. We present examples of [parallel compounding](#) (Section 5.1), and of phrasal (Section 5.2) and asymmetric translations (Section 5.3) of [compounds](#). A quantitative study on [cross-lingual compounding](#) and related experiments will be presented in the [Cross-lingual Compound Inspection \(XCI\)](#) in Section 10.2.

In Section 5.4, we discuss some [cross-lingual](#) indicators for [compound analysis](#): [compound identification](#) (5.4.1), [compound splitting](#) (5.4.2), [compound parsing](#) (5.4.3), the determination of [semantic indeterminacy](#) (5.4.4) and the prediction of the implicit [semantic relation](#) (5.4.5).

As main resource for this study, we use a subselection of the EUROPARL corpus, comprising ten European languages, which is described in more detail in Chapter 9.

## 5.1. Parallel Compounding

English	Danish	German	Dutch	Swedish
<i>climate change</i>	<i>klimændringerne</i>	<i>Klimawandel</i>	<i>klimaatverandering</i>	<i>klimatförändringarna</i>
<i>nuclear power</i>	<i>kernekræft</i>	<i>Kernenergie</i>	<i>kernenergie</i>	<i>kärnkraft</i>
<i>action plan</i>	<i>handlingsplan</i>	<i>Aktionsplan</i>	<i>actieplan</i>	<i>handlingsplan</i>
<i>euro area</i>	<i>eurområdet</i>	<i>Euroraum</i>	<i>eurozone</i>	<i>eurområdet</i>
<i>energy efficiency</i>	<i>energieffektivitet</i>	<i>Energieeffizienz</i>	<i>energie-efficiëntie</i>	<i>energieffektivitet</i>
<i>free trade</i>	<i>frihandel</i>	<i>Freihandel</i>	<i>vrijhandel</i>	<i>frihandel</i>

Table 5.1.: Examples of [parallel compounding](#)

Our first observations is that [compounding](#) is often realized in [parallel](#), i.e., there is a trend that a (lexicalized) [compound](#) is translated as a [compound](#). For example,

## 5. Cross-lingual Observations of Compounds

the English **compound** *death penalty* co-occurs with **closed compounds** in Danish (i.e., *dødsstraf*), in German (i.e., *Todesstrafe*), in Dutch (i.e., *doodstraf*) or in Swedish (i.e., *dödsstraff*). Some other examples of this **parallel compounding** is shown in Table 5.1.

The **compound** examples above mostly show frequent and lexicalized English **open compounds** composed of single nouns and their translations into four Germanic **closed compounding languages**. In cases of English **adjective-noun** constructions, the translations are more heterogeneous, as illustrated in Table 5.2.

English	Danish	German	Dutch	Swedish
<i>internal market</i>	<i>indre marked</i>	<i>Binnenmarkt</i>	<i>interne markt</i>	<i>inre marknaden</i>
<i>foreign policy</i>	<i>udenrigspolitik</i>	<i>Außenpolitik</i>	<i>buitenlands beleid</i>	<i>utrikespolitik</i>
<i>financial crisis</i>	<i>finanskrise</i>	<i>Finanzkrise</i>	<i>financiële crisis</i>	<i>finansiella krisen</i>

Table 5.2.: Examples of translations of adjective-noun sequences

For all examples and aligned languages in Table 5.1, we show the ratios of aligned one-**word equivalents** (based on automatic **word alignment**) in Table 5.3.

English	Danish	German	Dutch	Swedish
<i>climate change</i>	93.1%	90.2%	92.5%	97.1%
<i>nuclear power</i>	94.2%	93.8%	87.9%	95.2%
<i>action plan</i>	97.3%	90.6%	97.5%	92.8%
<i>euro area</i>	99.1%	99.1%	98.6%	99.1%
<i>energy efficiency</i>	92.7%	96.6%	96.8%	95.7%
<i>free trade</i>	79.6%	65.2%	71.0%	79.7%

Table 5.3.: Ratios of **parallel compounding**

For a comparison, we show the ratios of aligned one-**word equivalents** (based on automatic **word alignment**) for some more phrasal adjective-noun sequences (e.g., those which include deictic expressions) in Table 5.4.

English	Danish	German	Dutch	Swedish
<i>same time</i>	17.0%	21.2%	44.7%	70.5%
<i>last year</i>	1.6%	2.3%	0.8%	8.3%
<i>great deal</i>	18.9%	35.4%	11.2%	4.7%

Table 5.4.: Ratios of **parallel compounding** for adjective-noun phrases

The numbers illustrate that in contrast to **compounds** (Table 5.3), English phrases (Table 5.4) tend to be translated as **MWEs** in aligned **closed compounding languages**, i.e., as phrases.

## 5. Cross-lingual Observations of Compounds

For languages in which **compounding** is less prominent, most translations of **compounds** are phrasal (as will be discussed in Section 5.2), for example the French *changement climatique* ‘climate change’ (lit: ‘change climatic’), the Spanish *potencia nuclear* ‘nuclear power’ (lit: ‘power nuclear’), the Italian *piano d’azione* ‘action plan’ (lit: ‘plan of action’) or the Portuguese *área do euro* ‘euro area’ (lit: ‘area of euro’).

### 5.1.1. Parallel Closed Compounding

Another observation concerns some common **compounds** which are realized as **closed compounds** in many languages, even in those languages for which (**closed**) **compounding** is less prominent (e.g., Romance languages) and where **compounds** are usually translated into phrases, as discussed above. Most of the **cross-lingual equivalents** are cognates (e.g., the English *airport* and the Italian *aeroporto*). Table 5.5 shows some examples for *English*, for the **closed compounding languages** *Danish*, *German*, *Dutch* and *Swedish*, as well as for the Romance languages *Spanish*, *French*, *Italian* and *Portuguese*.










































English 	Equivalents in other European languages			
<i>airport</i>	<b>Danish</b> 	<b>German</b> 	<b>Dutch</b> 	<b>Swedish</b> 
	<i>lufthavnen</i>	<i>Flughafen</i>	<i>luchthaven</i>	<i>flygplats</i>
<i>airport</i>	<b>Spanish</b> 	<b>French</b> 	<b>Italian</b> 	<b>Portuguese</b> 
	<i>aeropuerto</i>	<i>aéroport</i>	<i>aeroporto</i>	<i>aeroporto</i>
<i>motorbikes</i>	<b>Danish</b> 	<b>German</b> 	<b>Dutch</b> 	<b>Swedish</b> 
	<i>motorcykler</i>	<i>Motorräder</i>	<i>motorfietsen</i>	<i>motorcyklar</i>
<i>motorbikes</i>	<b>Spanish</b> 	<b>French</b> 	<b>Italian</b> 	<b>Portuguese</b> 
	<i>motocicletas</i>	<i>motos</i>	<i>motocicli</i>	<i>motociclos</i>
<i>microphone</i>	<b>Danish</b> 	<b>German</b> 	<b>Dutch</b> 	<b>Swedish</b> 
	<i>mikrofon</i>	<i>Mikrofon</i>	<i>microfoon</i>	<i>mikrofon</i>
<i>microphone</i>	<b>Spanish</b> 	<b>French</b> 	<b>Italian</b> 	<b>Portuguese</b> 
	<i>micrófono</i>	<i>microphone</i>	<i>microfono</i>	<i>microfone</i>
<i>ecosystems</i>	<b>Danish</b> 	<b>German</b> 	<b>Dutch</b> 	<b>Swedish</b> 
	<i>økosystemer</i>	<i>Ökosysteme</i>	<i>ecosystemen</i>	<i>ekosystem</i>
<i>ecosystems</i>	<b>Spanish</b> 	<b>French</b> 	<b>Italian</b> 	<b>Portuguese</b> 
	<i>ecosistemas</i>	<i>écosystèmes</i>	<i>ecosistemi</i>	<i>ecossistemas</i>
<i>spokesperson</i>	<b>Danish</b> 	<b>German</b> 	<b>Dutch</b> 	<b>Swedish</b> 
	<i>talsmand</i>	<i>Sprecher</i>	<i>woordvoerder</i>	<i>talesman</i>
<i>spokesperson</i>	<b>Spanish</b> 	<b>French</b> 	<b>Italian</b> 	<b>Portuguese</b> 
	<i>portavoz</i>	<i>porte-parole</i>	<i>portavoce</i>	<i>porta-voz</i>

Table 5.5.: Examples for **parallel closed compounding**

## 5.2. Phrasal Translations

This observation is a complement to the previous observations about [parallel \(closed\) compounding](#), i.e., [compounds](#) that are realized as paraphrases ([phrasal equivalents](#)) in other aligned languages. As mentioned in Section 5.1, [phrasal equivalents](#) of [compounds](#) often occur in Romance languages. The most prominent pattern used for Romance paraphrases of English [binary nominal compounds](#) is the [complex nominal](#) (i.e., the [head](#) noun, a preposition and the [modifier](#) noun) and [noun adj](#) (i.e., the [head](#) noun and an adjective (mostly relational) denoting the [modifier](#)). For example, the [compound](#) *death penalty* can be translated to French as *peine de mort* (lit: ‘penalty of death’) or as *peine capitale* (lit: ‘penalty capital’). Table 5.6 shows some examples of English [kCs](#) and their [phrasal equivalents](#) in Romance languages.

English	Spanish	French	Italian
<i>death penalty</i>	<i>pena <b>de</b> muerte</i>	<i>peine capitale</i>	<i>pena <b>di</b> morte</i>
<i>developing countries</i>	<i>países <b>en</b> desarrollo</i>	<i>pays <b>en</b> développement</i>	<i>paesi <b>in</b> via di sviluppo</i>
<i>greenhouse gas</i>	<i>gases <b>de</b> efecto invernadero</i>	<i>gaz <b>à</b> effet de serre</i>	<i>gas <b>a</b> effetto serra</i>
<i>personal data</i>	<i>datos personales</i>	<i>données <b>à</b> caractère personnel</i>	<i>dati personali</i>
<i>motor vehicles</i>	<i>vehículos <b>de</b> motor</i>	<i>véhicules <b>à</b> moteur</i>	<i>veicoli <b>a</b> motore</i>
<i>part-time work</i>	<i>trabajo <b>a</b> tiempo parcial</i>	<i>travail <b>à</b> temps partiel</i>	<i>lavoro <b>a</b> tempo parziale</i>

Table 5.6.: Examples of phrasal equivalents in Romance languages

For all examples and aligned languages in Table 5.6, we show the ratios of aligned [multi-word equivalents](#) (based on automatic [word alignment](#)) in Table 5.7.

English	Spanish	French	Italian
<i>death penalty</i>	99.3%	98.1%	95.7%
<i>developing countries</i>	99.6%	98.5%	97.2%
<i>greenhouse gas</i>	100%	99.4%	99.0%
<i>personal data</i>	98.0%	98.3%	97.6%
<i>motor vehicles</i>	94.0%	80.4%	51.9%
<i>part-time work</i>	100%	100%	100%

Table 5.7.: Ratios of phrasal equivalents in Romance languages

The numbers prove that for Romance languages, most [compound](#) examples are realized in a [phrasal equivalent](#).

There are also [phrasal equivalents](#) in [closed compounding languages](#). For example, the [compound](#) *economic development* can be translated to *German* as *wirtschaftliche*

*Entwicklung*, the 3NC trade defence instruments can be translated to Swedish as *handelspolitiska skyddsinstrumenten* or the 4NC WTO market access agreement can be realized in Dutch as *WHO-overeenkomst over markttoegang*.

### 5.3. Asymmetric Translations

Sometimes, a compound is not literally realized in other languages, e.g., the semantic concept of a constituent (mostly the modifier) has changed. These non-literal translations are a challenge for cross-lingual compound analysis, as will be discussed in Section 5.4.

#### 5.3.1. Aspect Alternations

Cases of such alternations are shown for the language pair of English and German in Table 5.8, where mostly the modifier meaning has changed.

English	German
<i>highway</i>	<i>Autobahn</i> (lit: ‘car track’)
<i>air traveller</i>	<i>Flugreisender</i> (lit: ‘flight traveller’)
<i>airport</i>	<i>Flughafen</i> (lit: ‘flight port’)
<i>airline</i>	<i>Fluglinie</i> (lit: ‘flight line’)
<i>fresh start</i>	<i>Neuanfang</i> (lit: ‘new start’)
<i>dark side</i>	<i>Schattenseite</i> (lit: ‘shadow side’)
<i>pipe dreams</i>	<i>Wunschtraum</i> (lit: ‘desire dream’)
<i>dying words</i>	<i>letzte Worte</i> (lit: ‘last words’)
<i>bedroom</i>	<i>Schlafzimmer</i> (lit: ‘sleeping room’)
<i>health insurance</i>	<i>Krankenversicherung</i> (lit: ‘patient insurance’)
<i>Christmas tree</i>	<i>Christbaum</i> (lit: ‘Christ tree’)
<i>wheelchair</i>	<i>Rollstuhl</i> (lit: ‘rolling chair’)
<i>security camera</i>	<i>Überwachungskamera</i> (lit: ‘monitoring camera’)

Table 5.8.: Cross-lingual modifier alternations

While this phenomenon also occurs monolingually (e.g., *security camera* can be transformed to *monitoring camera*, or *superhighway* to *expressway* or *freeway* without (significantly) changing the meaning), we expect to see more alternations across languages. These alternations are partially regular (e.g.,  $airX \rightarrow Flug \tau(X)$ ). In most cases, the alternative modifier describes another aspect of the concept denoted by the compound

(e.g., a *flight* takes place in the *air*, the main purpose of a *bedroom* is *sleeping* or a *wheelchair* is *rolling*).

Another example of an aspect alternation both for **modifier** and **head** has already been shown in Table 5.5: the English **compound** *spokesperson* can be translated to German as *Wortführer* (lit: ‘word leader’), to Dutch as *woordvoerder* (lit: ‘word carrier’) or to French even as **exocentric compound** *porte-parole* (lit: ‘carry-words’).

### 5.3.2. Atomic Equivalentents

Another type of asymmetric translations are **compounds** that are lexicalized as an **atomic lexeme** in another language, for example the German *Handschuh* (lit: ‘hand shoe’) being translated to *English* as *glove*; or vice versa: the English **compound** *blackbird* being translated to *German* as *Amsel*.

### 5.3.3. Constituent Swapping

A final phenomenon of asymmetric translations is the case of **constituent swapping**, i.e., all **constituents** are translated literally, but the translation of the **head** functions as **modifier** and the translation of the **modifier** functions as **head**.

For example, the German **compound** *Perlzucker* ‘pearl sugar’ is literally and symmetric translated to Dutch as *parelsuiker* (lit: ‘pearl sugar’) but to French as *perles de sucre* (lit: ‘pearls of sugar’). Here, the German/English/Dutch **head** *sugar* has become **modifier** in the French **complex nominal**, whereas the German/English/Dutch **modifier** *pearl* has become **head** in French. A possible explanation of this case of **constituent swapping** is the **compound** class. *Pearl sugar* can be considered as being both *pearls* and *sugar*, i.e., an appositional **compound** (as described in Section 3.7.4).

Certainly, this is also a monolingual phenomenon (*Zuckerperlen* ‘sugar pearls’ vs. *Perlzucker* ‘pearl sugar’), but it should be noted that cases of **constituent swapping** can occur across languages and pose another challenge for **cross-lingual** methods for **compound analysis**.

Another monolingual and **cross-lingual** type of **constituent swapping** is found with **compounds** composed of a noun sequence and a nominalized adjective as **head**, which refers to a property of the **modifier**. For example, the English **compound** *exchange rate stability* can be translated to the Dutch paraphrase *stabiele wisselkoersen* ‘stable exchange rate’. For this type, the reference of the property denoted by the **head** is important. While the **constituents** can be swapped for *Strukturfestigkeit* ‘structural



strength’ to *feste Struktur* ‘strong structure’ (i.e., the **head** refers to a property of the **modifier**), it is not possible for *Stoßfestigkeit* ‘shock resistance’, because the property denoted by the **head** usually does not refer to the **modifier** (i.e., \**fester Stoß* ‘strong hit’).

### 5.4. Cross-lingual Indicators for Compound Analysis

#### 5.4.1. Compound Identification

The trend of **parallel (closed) compounding** (described in Section 5.1) can be used as indicator for the task of **compound identification**: if an English **word** sequence is translated into **compounds** in aligned (**closed compounding**) languages, it has a high chance of being a **compound** itself. We will investigate the potential of this kind of **cross-lingual** evidence in Part C of this thesis.

#### 5.4.2. Compound Splitting

The scenario of having a multiword **equivalent** aligned to a **closed compound** provides a beneficial indicator for **compound splitting**, i.e., for determining the composed **constituents**, as shown in previous work (Brown, 2002, Koehn and Knight, 2003). For example, the German **closed compound** *Menschenrechte* can be **split** by using the English **equivalent** *human rights* and a suitable method for mapping the English **constituents** onto substrings of the German **closed compound** using a bilingual resource. For example, *human* is listed as translation for *Mensch* and *rights* as translation for *Rechte*. Tolerant string matching (considering **constituent inflection** and lowercased **heads**) yield the correct **split point** *Menschen|rechte*.

#### 5.4.3. Compound Parsing

As discussed in Section 3.6.4, complex **compounds** that have three or more **constituents** are structurally ambiguous, e.g., *plastic water bottle* could be LEFT- or RIGHT-branched, usually expressing different meaning. Such complex **compounds** can be translated to paraphrases, as described in Section 5.2. Some of these **phrasal equivalents** can reveal the **internal structure** of a **kC**. For example, the **4NC** *energy efficiency action plans*, that has five possible structures (cf. Table 3.2), can be **parsed** using the German **equivalent** *Aktionspläne für Energieeffizienz* (lit: ‘action plans for energy efficiency’) with the re-

spective [word](#) alignments: this [phrasal equivalent](#) points to a balanced tree structure, shown in Figure 5.1.

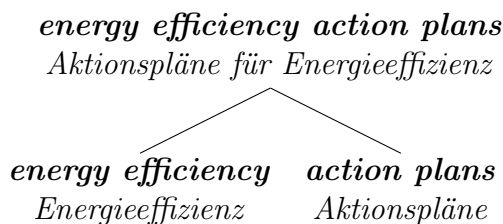


Figure 5.1.: Example of a balanced tree structure

We will further investigate this type of [cross-lingual compound](#) structure evidence in more detail and will develop different [compound parsing](#) methods in Part E of this thesis.

#### 5.4.4. Prediction of Semantic Indeterminacy

In Section 3.8.3, we discussed the phenomenon of [semantic indeterminacy](#), e.g., a [3NC](#), such as *college football player*, being semantically equivalent with respect to both a LEFT- and a RIGHT-branched structure.

As described in Section 5.2 and Section 5.4.3, [phrasal equivalents](#) of [kCs](#) can provide evidence for a certain syntactic structure, as illustrated in Figure 5.1. With respect to these [cross-lingual](#) indicators for [compound parsing](#), we observed [parallel](#) inconsistencies for some [compound tokens](#), e.g., evidence for both a LEFT- and a RIGHT-branched structure of a certain instance of a [3NC](#), i.e., for [semantic indeterminacy](#).

While the English [3NC](#) *tobacco advertising ban* is realized in German as *Werbeverbot für Tabakerzeugnisse* (lit: ‘advertising ban for tobacco products’) (providing a RIGHT-branched reading), the Danish [equivalent](#) is *forbuddet mod tobaksreklamer* (lit: ‘ban of tobacco advertising’) (providing a LEFT-branched reading). Similarly, the English [3NC](#) *animal welfare standards* is once realized in Dutch as *normen op het gebied van dierenwelzijn* (lit: ‘standards in the field of animal welfare’) (i.e., a LEFT-branched reading) and in German as *Wohlfahrtsstandards für Tiere* (lit: ‘welfare standards for animals’) (i.e., a RIGHT-branched reading). We can also find evidence for [semantic indeterminacy](#) for longer [compounds](#). For example, the [4NC](#) *book price fixing schemes* shows a strong variety across the languages. It is realized in Danish as *fastprisordningerne for bøger* (lit: ‘price fixing schemes for books’), in Dutch as *vaste boekenprijisregelingen* (lit: ‘fixed book price schemes’) and in French as *règlements concernant les prix fixes du livre* (lit: ‘schemes concerning the fixed price of books’).

We will further investigate this kind of [cross-lingual](#) evidence for [semantic indeterminacy](#) within the task of [cross-lingual compound parsing](#) in Part E of this thesis.

### 5.4.5. Prediction of Semantic Relations

Apart from [constituent equivalents](#), [phrasal equivalents](#) also include additional material such as [function words](#) (e.g., prepositions) that connects the [constituent equivalents](#), as shown for Romance languages in Table 5.6. As has been observed in previous work on semantic interpretation of [compounds](#), the implicit [semantic relation](#) holding between [modifier](#) and [head](#) seems to correlate with the preposition used in [phrasal equivalents](#) in Romance languages, e.g., *chocolate cake* (i.e., a cake `made_with` chocolate) is realized in French as *gâteau au chocolat*, whereas *wedding cake* (i.e., a cake `made_for` a wedding) is realized as *gâteau de mariage* (Ziering and Van der Plas, 2014). Girju (2007) proved the potential of the knowledge about the prepositions of [phrasal equivalents](#) in five Romance languages, encoded as features in a [Support Vector Machine \(SVM\)](#), for the automatic classification of [semantic relations](#) in [compounds](#). Celli and Nissim (2009) showed that prepositions in Italian [complex nominals](#) (of the form `noun prep noun`) are beneficial features in a supervised [semantic relation](#) classifier.

English	French	made_of?
<i>glass bottle</i>	<i>bouteille <u>en</u> verre</i>	✓
<i>glass jar</i>	<i>pot <u>en</u> verre</i>	✓
<i>leather jacket</i>	<i>vest <u>en</u> cuir</i>	✓
<i>paper basket</i>	<i>corbeille <u>à</u> papier</i>	✗
<i>Iron curtain</i>	<i>rideau <u>de</u> fer</i>	✗

Table 5.9.: Examples of French [phrasal equivalents](#) for `made_of` [compounds](#)

Actually, many English [compounds](#) having a `made_of` relation are realized in French with the preposition *en* as shown in Table 5.9. In the case of other [semantic relations](#) (e.g., the `container` relation as in *paper basket*) or even in metaphorical readings (e.g., *Iron curtain*), an alternative preposition is used.

Certainly, Romance prepositions cannot fully correlate with all possible [semantic relations](#) of [compounds](#) (and even more so, given that there are virtually infinitely many relations possible, as mentioned in Section 3.8.2), because there are only a few prepositions used in [phrasal equivalents](#) (e.g., the most frequent French prepositions having such a function are *de*, *à*, *en* and *sur*). Moreover, most Romance languages have a

## 5. Cross-lingual Observations of Compounds

universally applicable filler (e.g., *de* in French) which can substitute a more specific preposition. For example, the **compound** *leather shoes* (i.e., shoes *made out of* leather) can be realized in French as *chaussures en cuir* or as *chaussures de cuir*. Actually, we find both translations with equal frequency in EUROPARL. For *glass bottle*, we find even more instances of *bouteille de verre* than of *bouteille en verre*.

The bottom line is that while not sufficiently expressive for modelling any **semantic relation**, Romance prepositions (extracted from **phrasal equivalents**) can be used as an additional feature for learning the **semantic relation** (Celli and Nissim, 2009, Girju, 2007).

# 6. Bottom Line of Nature of Compounds

## 6.1. Summary

This Part B outlined the nature of the main subject of this thesis, **compounds**, i.e., it described general properties of **compounds** and their **constituents**, and the definition of **compounds**. Moreover, it gave a **multilingual** outlook on **compounding** in several relevant languages.

In the **introduction chapter**, **Chapter 2**, we firstly introduced a basic description of **compounds** (2.1) used for the remainder of this part. Moreover, we provided an outline of the content of this part (2.2).

In **Chapter 3**, we described various **general aspects of compounds**. Besides **compounds**, we outlined other types of **MWEs** (3.1). Some naming conventions about different categories of **compounds** were described in Section 3.2. The productivity of **compounds** was explained in Section 3.3. Some possible functions of **compounding** were discussed in Section 3.4. Then, we described the different spelling forms (**open**, **closed**, **hyphenated**, mixed) of **nominal compounds** (3.5), and the different **constituent types** (3.6) and **compound classes** (3.7). The semantics of **compounds** was outlined in Section 3.8: the compositionality of **compounds** (3.8.1), the implicit **semantic relation** between **modifier** and **head** (3.8.2) and the **semantic indeterminacy** (3.8.3). Finally, we discussed **compounding** for different **target languages** (3.9) that will be relevant in experiments in the subsequent parts of the thesis: English (3.9.1), German (3.9.2), Dutch (3.9.3) and Afrikaans (3.9.4).

The **controversy found in discussions on the definition** of **compounds** was presented in **Chapter 4** and **linguistic criteria** for **compoundhood** were presented: orthographical (4.3), morphological (4.4), phonetic/prosodic (4.5), syntactic (4.6) and semantic (4.7) **criteria**.

In the previous chapter, **Chapter 5**, we regarded **compounding from a cross-**

**lingual point of view**, i.e., how **compounds** are translated (e.g., to **compounds** (5.1) or paraphrases (5.2)). We presented some cases of asymmetric **compound** translations (5.3), such as **constituent swapping** (5.3.3). Moreover, we discussed **cross-lingual** indicators for various **compound analysis** tasks, including **compound identification** (5.4.1), **compound splitting** (5.4.2) and **compound parsing** (5.4.3).

Finally, in this **Chapter 6**, we summarize and conclude.

## 6.2. Conclusion

### Possible Conclusions in Linguistics Literature

As a final result of the **compound** definition problem (outlined in Chapter 4), Lieber and Štekauer (2009) argue that there is no fully reliable **criterion** for defining **compounds** and distinguishing them from phrases or **atomic lexemes**.

They discuss different possible conclusions that can be drawn from this:

- (1) there is **no compounding word formation** at all, (Marchand, 1967, Spencer, 2003)
- (2) there are no clear classes ‘**compound**’ and ‘non-compound’, but instances of **more or less compoundlike expressions** (Lieber and Štekauer, 2009)
- (3) all **noun+noun** constructions can be considered as **compounds** (Olsen, 2000)
- (4) we can remain **agnostic** on whether there is a distinction between **compounds** and their corresponding phrases (Plag, 2006)

### Grounding for this Thesis

We follow the second conclusion, saying that there is no clear **compound** class but there are constructions which can be considered more or less compoundlike, i.e., “compounding is a gradient, rather than a categorical phenomenon, with prototypical examples and fuzzy edges” (Lieber and Štekauer, 2009).

The more **linguistic criteria** for **compoundhood** are met, the more compoundlike is the underlying expression. This perspective on **compounds** will be the grounding for the subsequent parts of this thesis. In a pilot study in Chapter 10, the **Linguistic Criterion Inspection (LCI)**, human annotators will rate the degree of **compoundhood** according to this grounding (10.1).

## 6.3. Motivation and Outlook

### Cross-linguality

We presented and discussed several [cross-lingual](#) patterns of [compounding](#) observed in a [parallel corpus](#) in Chapter 5. Some of these observations motivated us to dig deeper into [parallel data](#), and answer questions such as what factors lead to what types of [compound](#) translations, what regularities can be observed for asymmetric translations, etc. Moreover, the possible indicators for [compound analysis](#) presented in Section 5.4 motivated us to develop [cross-lingual](#) methods for [compound analysis](#).

In Part C, we will exploit [cross-lingual](#) indicators for the [compound identification](#) (as described in Section 5.4.1) as a type of [cross-lingual supervision](#). In Part E, we will exploit [cross-lingual](#) evidence for [compound parsing](#) (as discussed in Section 5.4.3) and for the prediction of [semantic indeterminacy](#) (as presented in Section 5.4.4), and will develop [cross-lingually supervised](#) methods.

### Multilingual Compound Database

After all observations and insights about monolingual, [multilingual](#) and [cross-lingual compounding](#), we felt that a [multilingual](#) database of [compounds](#) would be a very useful resource for the community and beyond. It can serve as resource for [NLP](#) tools exploiting [cross-lingual](#) indicators (e.g., [compound splitting](#)) and for further empirical experiments and linguistic research about the nature of [compounds](#): what given [linguistic criteria](#) work best for defining and [identifying compounds](#), and can we find novel regularities from observed [compounds](#)? These questions will be addressed in the following part, Part C.

*6. Bottom Line of Nature of Compounds*



Part C.

## Compound Identification



# 7. Introduction to Compound Identification

In this part, we present and elaborate parts of the work published in Ziering and Van der Plas (2014).

After having discussed the nature of **compounds**, as given in linguistics literature, and the controversy of defining **compounds** in Part B, we address the **identification of compounds** in this part. For **identifying compounds**, we first have to determine a concept of **compoundhood** that serves as foundation for the **identification** process. We aim to find suitable features for an automatic **identification** method. To this end, we perform some pilot studies. As a final result of **compound identification**, we create a database composed of English **nominal compounds** and their **cross-lingual equivalents** in various European languages, extracted from a **parallel corpus**.

While there are various **word** categories possible for the English **compound head** (as discussed in Section 3.9.1), in the scope of this thesis, we decided to restrict to the most frequent **head** category, **nouns**, i.e., **nominal compounds**. Alternative less frequent **head** categories (e.g., verbs - **verbal compounds**) will be addressed in future work.

## 7.1. Motivation

In the previous part, we talked about the nature of **compounds**, about the controversy of the **compoundhood** definition (Chapter 4) and about some **cross-lingual** observations (Chapter 5) we made when investigating **compounding** in a **parallel corpus**.

The motivation is divided into two subsections. The first subsection (7.1.1) concerns the motivations for performing pilot studies on the definition of **compounds** (presented in Chapter 10) and for developing a **compound identification** method (presented in Chapter 11). The second subsection (7.1.2) describes the motivations for building a **multi-lingual compound** resource (e.g., the **Europarl Nominal Compound Database (ENCD)**, presented in Chapter 12).

### 7.1.1. Motivation for Compound Identification

#### The Lack of Definition of Compounds in NLP

Linguistics literature discuss the definition of **compounds** controversially and propose some more or less reliable **linguistic criteria**, for which there are counterexamples. In the field of **NLP**, previous work on **compound analysis** mostly avoids to tackle the issues of **compound** definition. Instead, commonly non-debatable cases of **nominal compounds** (e.g., **2NCs**) were extracted and analyzed.

This motivated us to get an idea of the importance and relevance of some **linguistic criteria** for the determination of the **compoundhood status** of an expression. Moreover, we want to get an impression about the agreement of the validity of **linguistic criteria**, i.e., how much (divergent) subjectivity influences the judgement of these **criteria**?

Finally, we want to develop a working definition of **compounds** that can be used for future methods for **compound identification**, irrespective of the **compound** class.

#### Identification Evaluation

In order to judge the performance of our **compound identification** method, there is need for a gold standard that provides **compound** annotations in context. Moreover, the **Inter-Annotator Agreement (IAA)** for the **compound** annotation task provides an upper bound for the **identification** task.

#### Importance of Compound Identification in NLP

Knowing about the **compoundhood status** of a **target** expression is inevitable for many **NLP** tasks. For example, an **SMT** system has to know about the **compoundhood status** of *French teacher*, before translating it to German (**compound**: *Französischlehrer*, phrase: *französischer Lehrer*), or for **NLU**, we have to know whether *friendship* is a certain type of *ship*.

### 7.1.2. Motivation for Multilingual Compound Resource

#### Research on Cross-lingual Equivalents

Some of the observations made in previous chapters motivated us to have a deeper look into the **parallel** correspondences of **compounds**, as occurring in **parallel data**. We observed different types of **compound** translations (e.g., an **open** or **closed compound** being

translated to a **closed compound** or to a paraphrase), discussed in Sections 5.1 and 5.2. Research on **cross-lingual compounding** could provide evidence for a correlation between the attributes of a **target compound** and the **compoundhood status** of **cross-lingual equivalents** (e.g., how does the semantics of an English **compound** correlates with the way it is translated to German). Another observation concerned asymmetric **compound** translations (e.g., *airport* being translated to German as *Flughafen* (lit: ‘flight port’)) (5.3). Using bilingual dictionaries, such asymmetric translations can be recognized. One might find some regularities for **compound** translation types or asymmetric translations, which could be helpful in **Natural Language Generation (NLG)**, e.g., as part of an **SMT** system. Therefore, a **multilingual compound database** can serve as resource for a deep research in the field of **cross-lingual equivalents** of English **compounds**.

### Training Data for Cross-lingual Compound Analysis

In Section 5.2, we discussed some possible ways of exploiting phrasal **compound** translations as an indicator for different **NLP** tasks concerning **compound analysis** (e.g., for **compound splitting**, for **compound parsing** or for the **semantic relation** determination). Therefore, a **multilingual compound database** can be used for finding promising features for **cross-lingually supervised** methods for **compound analysis**. Moreover, such a resource could serve as training data for the respective **NLP** task (e.g., learning the probability of Romance prepositions for a given **semantic relation**). We will show in the remainder of the thesis that the database is used for many of our approaches to **compound analysis**.

### Support for Compound Definition

The definition of **compounds** is discussed highly controversially in linguistics literature, and some more or less reliable but commonly established **linguistic criteria** have been proposed.

While we will exploit **linguistic criteria** for finding a practical approximation of the **compound** definition, serving as grounding for our **compound identification** method, a **multilingual compound database** can be a suitable resource for further empirical experiments and linguistic research about the definition and nature of **compounds**. The resource can help for observing regularities pointing to novel (possibly **cross-lingual**) **linguistic criteria**.

## 7.2. Contributions and related Research Questions

Besides minor observations made during data analysis and experiments (e.g., error analyses), we claim to provide the following contributions along with this thesis part. Moreover, in this section, we repeat and refine some research questions posed in Section 1.3.

### 7.2.1. New Insights about the Notion of Compoundhood

The first contribution of this part represents a main contribution of this thesis, as discussed in Section 1.4.1.

Linguistics literature discuss the definition of **compounds** controversially and propose some more or less reliable **linguistic criteria**. For all of these **criteria**, there are counterexamples. In the field of **NLP**, previous work on **compound analysis** mostly avoids to tackle the issues of **compound** definition. Instead, commonly non-debatable cases of **nominal compounds** (e.g., German **closed compounds** or sequences of two English nouns, i.e., **binary noun compounds**) were extracted and analyzed. To the best of our knowledge, there is no previous work in computational linguistics that addresses the definition of **compounds**, as well as the automatic **identification** of such a large number of different **nominal compound** classes.

Before **identifying compounds**, we perform two pilot studies that will help us find a suitable definition of **compounds** that is implementable for an automatic **identifier**.

The first pilot study is a **Linguistic Criterion Inspection (LCI)**. We decided to inspect corpora to determine the suitability of **criteria** proposed by linguistics literature. In an experiment, we asked human annotators to **identify** potential **nominal compounds**, rate their degree of **compoundhood** and inspect the applicability of various **linguistic criteria** on these examples.

There are two main insights that this part of the thesis contributes: (1) insights on the definition of **compounds** and (2) insights on the **cross-linguality**.

**Insights for the Definition.** In the **LCI**, we make use of human ratings of **linguistic criteria** for different kinds of **nominal compound** candidates. We show which of the commonly established **linguistic criteria** are most reliable for the definition of **compounds** (Section 10.1).

We aim to answer the following research (sub)questions.

**RQ\_1-A:** What **linguistic criteria** help to **identify compounds**?

**RQ\_1-A-i:** Which **linguistic criteria** show highest and lowest **IAA**?

**RQ\_1-A-ii:** What is the [identification](#) agreement, serving as UPPER bound for our [compound identification](#) method?

**RQ\_1-A-iii:** What is the agreement for rating [compoundhood](#)?

**Cross-lingual Insights.** In the second pilot study, we look at [compounding](#) from a [cross-lingual](#) perspective, as given by a [parallel corpus](#). The [Cross-lingual Compound Inspection \(XCI\)](#) provides an overview how English [2NCs](#) (as given by an external gold standard) are mostly formed in other languages. In the [XCI](#), we present a quantitative study of how [cross-lingual equivalents](#) of English [2NCs](#) are formed (Section 10.2).

We aim to answer the following research question.

**RQ\_1-B:** What are the most frequent formations of [cross-lingual equivalents](#) for an English [compound](#)?

Our research and experiments shed new light on the notion of [compoundhood](#) and the [cross-lingual](#) behavior of [compounds](#).

### 7.2.2. Cross-lingual Compound Identifier

One of our observations during the [XCI](#) pilot study is that English [2NCs](#) are frequently realized as [closed compounds](#) or [atomic words](#) in [parallel closed compounding languages](#). Exploiting this regularity, we will develop a [cross-lingual compound identification](#) method that relies on knowledge about the alignments to [equivalents](#) spelled as one [word](#). This [compound identification](#) method constitutes the second contribution of this part, which will be presented in Chapter 11. This language-independent method is applicable to any [parallel corpus](#), in which a [target language](#) (e.g., *English*) is aligned to a set of expressive [support languages](#) (e.g., [closed compounding languages](#)). As will be shown in an experiment, the restriction to English candidates that are aligned to [closed compounds](#) is a beneficial [criterion](#) for a [compound](#) resource with high precision.

We aim to answer the following research (sub)questions.

**RQ\_1-C:** Is [cross-lingual](#) information beneficial for the automatic [identification](#) of [compounds](#) in context?

**RQ\_1-C-i:** What are the limitations of the use of [cross-lingual](#) evidence for [compound identification](#)?

### 7.2.3. Lexical Resources

The third contribution of this part concerns lexical resources of [nominal compounds](#), as discussed in Section [1.4.4](#).

#### Europarl Nominal Compoundhood Ratings

For the [LCI](#) and for the evaluation of our [identifier](#), two native English-speaking experts annotated [nominal compounds](#) with ratings about their [compoundhood](#) and the validity of some [linguistic criteria](#) in a set of EUROPARL<sup>1</sup> sentences. More details about the [Europarl Nominal Compoundhood Ratings \(ENCR\)](#) will be described in Section [10.1.1](#).

#### Europarl Nominal Compound Database

The [compound identification](#) method presented in this part is applied to EUROPARL. The result, the [Europarl Nominal Compound Database \(ENCD\)](#), is a [compound](#) resource with English [nominal compounds](#) of any [compound size](#) (in terms of [atomic constituents](#)) and their [cross-lingual equivalents](#). Besides the [word forms](#), the [ENCD](#) contains information about [lemmas](#), [PoS](#), [split points](#), etc. More details about the [ENCD](#) will follow in Chapter [12](#). This database can serve various purposes, as described in Section [7.1](#).

## 7.3. Outline

The [compound identification](#) part is structured in the following way. In Chapter [8](#), we will have a look at related and previous work on the [identification](#) and discovery of [compounds](#) ([8.1](#)), and at previous work on [compound](#) resources ([8.2](#)). In Chapter [9](#), we will describe the [parallel corpus](#) that forms the grounding source of most experiments on [cross-lingually supervised](#) methods for [compound analysis](#) within this thesis. In Chapter [10](#), the two pilot studies ([LCI](#) and [XCI](#)) will be presented. The main [cross-lingual compound identification](#) method will be explained in Chapter [11](#). The result of our [identifier](#) applied to EUROPARL, the [ENCD](#), will be outlined in Chapter [12](#). The quality of the [ENCD](#) (and therewith of the [cross-lingual identification](#) method) is evaluated in an experiment described in Chapter [13](#). Finally, Chapter [14](#) summarizes and concludes this thesis part on [cross-lingual compound identification](#), Part [C](#).

---

<sup>1</sup>[statmt.org/europarl](http://statmt.org/europarl)



# 8. Related Work on Compound Identification

In this chapter, we present an outline of previous related work on the subjects of this part, i.e., the **identification** and **discovery of compounds** (8.1), and **compound resources** (8.2).

## 8.1. Methods for the Identification and Discovery of Compounds

In this section, we present the most relevant previous related work that address the manual, automatic or semi-automatic **identification** and **discovery of compounds** and related expressions (e.g., **MWEs**). In the description of each approach, we focus on **six methodological features**:

1. **Compound class** - Is the method designed for a specific kind of **compound** (e.g., **closed** or **open compounds**, **nominal compounds** or **noun compounds**) with a specific arity (e.g., **two-Noun Compounds (2NCs)**, **three-Noun Compounds (3NCs)**, ...) or is it applicable for finding any type of **MWE**?

Our **compound identification** method, which will be presented in Chapter 11, is designed for both **open** and **closed nominal compounds**. However, it is plausible to apply our method to alternative **PoS** categories of **compounds** that have a similar **cross-lingual** behaviour, e.g., **adjectival compounds** such as *bullet proof* aligned to the German *kugelsicher* or the Dutch *kogelvrij*.

2. **Language** - Which language(s) is the method designed for?

Our **compound identification** method is designed to be language-independent. We distinguish between the **target language** (i.e., the **compound's** language) and the **support languages** (i.e., the languages aligned to the **target compound**). We

will exemplify our **identification** method by applying it to the **parallel** EUROPARL corpus (Chapter 9). As result, we get the **Europarl Nominal Compound Database (ENCD)**, in which *English* is the **target language** and the four **closed compounding languages** *Danish, German, Dutch* and *Swedish* are the aligned **support languages**. In many cases, an English **nominal compound** is translated to a **closed compound** in these languages (as discussed in Section 5.1). Thus, our **identification** method can be considered as a **multilingual** method to some extent, because there are **identified compounds** on both the **target language** side and the **support languages**’ side.

3. **Contextuality** - Is the method an **identification** method, which highlights the **compounds** in context based on a predefined lexicon, or is it a **discovery** method, that creates such a lexicon with out-of-context **compound** types (Constant et al., 2017)? As discussed in Section 1.2.1, there are expressions that vary between a **compound** and a phrasal reading depending on context, e.g., *French teacher* as a person teaching the school subject ‘French’ and as a teacher who is French. Thus, providing a context for **compound** candidates is informative.

Our proposed method **discovers nominal compound tokens** in context. Although the method does not annotate a corpus and does not rely on a lexicon, it provides information about the exact position of the **discovered compound tokens** and extracts the surrounding sentence as context. Therefore, we consider our method, proposed in Chapter 11, as an **identification** method.

4. **Human support** - Is the method performed manually (i.e., with full human support), automatically (i.e., machine-based without any human support) or semi-automatically (i.e., machine-based with only partial human support, e.g., in a post-filtering step).

Our method is fully automatic.

5. **Supervision** - Is the method supervised (i.e., based on **compound**-annotated training data) or unsupervised (e.g., based on bigram corpus frequency)?

Our **compound identification** method is unsupervised because it does not rely on **nominal compound** annotations. However, it exploits the **cross-lingual** information about **compounding** (in terms of aligned **closed compounds**) provided with **parallel corpora**. In this sense, our method can be considered as being based on **cross-lingual supervision**, a subtype of **indirect supervision**.

6. **Features** - Which features are used for **identifying** or **discovering compounds** (e.g., **PoS Patterns** or corpus frequency)?

In our **compound identification** method, we defined two types of features: **candidate features** and **cross-lingual features**. The candidate features are used for selecting **compound** candidates (which will be described in Section 11.1). For this purpose, we used a set of predefined **PoS patterns**. Using **PoS patterns** for extracting **MWEs** is a common approach for most previous work (e.g., the **mwetoolkit** (Ramisch et al., 2010c)). The **cross-lingual** features are used for the **cross-lingual** validation (which will be described in Section 11.3). These features are motivated by the **cross-lingual** observations about **compounding**, outlined in Chapter 5, more specifically, the **parallel compounding** (5.1). We will use the degree of **closed compounds** among the **cross-lingual equivalents** of a **target compound**.

For structuring Section 8.1, we group previous work with respect to the **compound class** feature.

### 8.1.1. Two-Noun Compounds

In most cases, previous work deal with the easiest **compound** class, **2NCs**.

**Lauer (1995b)** used an automatic and unsupervised heuristic for the discovery of English **2NCs**.

Lauer (1995b) defined the set of noun pairs which are not in the context of another noun, as given in Formula 8.1, where  $N$  is the set of all nouns.

$$C = \{(w_2, w_3) | w_1 w_2 w_3 w_4; w_1, w_4 \notin N; w_2, w_4 \in N\} \quad (8.1)$$

Lauer (1995b) applied his heuristic on the Grolier Multimedia Encyclopedia with a set of 90,000 unambiguous nouns and reports a **discovery** accuracy of 97.9%.

**Lapata and Lascarides (2003)** present a method for distinguishing infrequent **compounds** from non-**compounds** (i.e., nonce terms), where statistical **Association Measures (AMs)** do not work (e.g., for hapax legomena). In the presented experiments, Lapata and Lascarides (2003) restrict to English **2NCs**. They adapted the heuristics of Lauer (1995b) but used the **PoS**-tagged and **lemmatized** British National Corpus (BNC: (Burnard, 2000)) and the chart parser Gsearch (Corley et al., 2001) for sampling coherent noun sequences.

The proposed method is an automatic supervised **identification** approach, which includes the context as indicating feature for the **identification** of **2NCs**.

## 8. Related Work on Compound Identification

As features for the distinction between 2NCs and non-compounds, Lapata and Lascarides (2003) utilized statistical features such as constituent frequency, constituent type probability (e.g., how likely can a given noun be used as modifier) or the frequency of semantic concept pairs (i.e., the constituents are generalized to a set of semantic classes using WordNet). For avoiding incoherent noun sequences such as *may push* in *Their different responsibilities in relation to the public may push them in opposite directions*, the context (e.g., a succeeding pronoun) can indicate a false extraction. Lapata and Lascarides (2003) encodes the context of a potential 2NC as PoS tags of the four preceding and succeeding words.

In an experiment, two human annotators identified 1000 hapax legomenon 2NCs from the BNC, which were used in a 10-fold cross-validation with a C4.5 decision tree learner (Quinlan, 1993) and a Naive Bayes classifier (Duda and Hart, 1973). The untrained annotators were instructed with annotation guidelines, leading to a Kappa score of 0.80 (Cohen, 1960) and an agreement rate of 89%.

Ramisch et al. (2010b) present a case study for discovering English 2NCs in Europarl using the `mwetoolkit` (Ramisch et al., 2010c), presented in Section 8.1.5. After preprocessing the English part of Europarl, noun-noun sequences with a minimum frequency of 2 are extracted. As source of frequency for unigrams and bigrams, Ramisch et al. (2010b) used the Europarl corpus and search engine hits from Google<sup>1</sup> and Yahoo!<sup>2</sup>. For each 2NC candidate, four AMs are calculated and 2NCs below a predefined threshold are discarded.

Ramisch et al. (2010d) investigated the impact of techniques for combining heterogeneous corpora (aiming to minimize the negative effects of data sparseness on the performance of empirical NLP methods) on the discovery of English 2NCs. They extracted 2NCs from the general-purpose Europarl corpus and from the specialized (biomedical) Genia corpus (Ohta et al., 2002) using the approach of generating candidates using PoS patterns and filtering them using statistical AMs (Evert and Krenn, 2005, Pecina, 2008, Ramisch et al., 2010c). Ramisch et al. (2010d) came to the conclusion that counts from the web or from combined corpora cannot help to extract specialized 2NCs, because these counts “do not help minimize data sparseness”. In contrast, the extraction of general-purpose 2NCs can be improved using web-based counts (Ramisch et al., 2010d).

Ivanova and Wehrli (2015) developed a compound identification method that is based on syntactic analysis and lexical information. Their method uses two linguistic

---

<sup>1</sup>google.com

<sup>2</sup>yahoo.com

criteria for compoundhood: the **inseparability criterion** (4.6.1) and the **inability to modify the modifier** (4.6.2). These **linguistic criteria** are validated using the syntactic structure, given by the output of the FIPS parser (Wehrli, 2007). As input, Ivanova and Wehrli (2015) used the output of a **Speech Recognition (SR)** system.

### 8.1.2. Nominal Compounds

Vincze et al. (2011) presented two automatic approaches based on Wikipedia (“dictionary-based”) and on a **Conditional Random Fields (CRF)** classifier (“machine learning based”) for **identifying nominal compounds** and named entities in context. While their experiments are conducted for English, the approaches seem to be easily adaptable to other Wikipedia languages.

Vincze et al. (2011) developed an unsupervised Wikipedia-based approach, for which they compiled a list of Wikipedia-internal links (with a text of 2-4 lowercased tokens) and their link frequency. They described four **identification** principles:

**Match:** a **word** sequence is a **nominal compound** if it occurs in the link list with a frequency above a predefined threshold

**Merge:** if A B and B C are valid **nominal compounds**, then A B C is also considered as valid **ternary nominal compound**

**PoS rules:** if the **word** sequence matches with one of various predefined **PoS Patterns** (e.g., ADJ + NOUN) and has a frequency above a predefined threshold, it is considered as a **nominal compound**

**Combined:** a combination of all three principles

Moreover, Vincze et al. (2011) developed both a supervised and an unsupervised machine learning based **identification** approach. They employed a **CRF** model with a set of features defined by Szarvas et al. (2006) for the task of **multilingual NER**, including orthographical features (e.g., capitalization, **word** length, ...), dictionary matches and information about frequency, **PoS** and context. The **CRF** model was trained and tested with the Wiki50 corpus, described in Section 8.2.5. As an alternative training set, Vincze et al. (2011) created a silver standard by applying the **combined** Wikipedia-based **identifier** to a random set of 5000 Wikipedia articles.

Nagy T. et al. (2011) used the **nominal compound identifier** of Vincze et al. (2011) for showing the usability in keyphrase extraction.

**Nagy T. and Vincze (2013)** conducted some experiments on the methods of Vincze et al. (2011). They showed that the size of the underlying Wikipedia corpus has an impact on the quality of the Wikipedia-based approach (i.e., the performance, in particular recall, is improved for larger corpora) and that the size of the automatically generated silver standard influences the [CRF-based identifier](#).

### 8.1.3. Bigrams including Nominal Compounds

**Keller et al. (2002)** developed a method for getting web frequencies for [adjective-noun](#) bigrams (including [nominal compounds](#)), [noun-noun](#) bigrams (including [noun compounds](#)) and [verb-object](#) pairs. For the bigram sampling, Keller et al. (2002) followed the approach of Lapata et al. (1999) and Lapata et al. (2001), who constructed known and unknown [adjective-noun](#) bigrams from the BNC.

While the experiments of Keller et al. (2002) are conducted on English bigrams, it is plausible to consider this approach language-independent.

The method automatically [discovers](#) out-of-context (or even unattested) bigrams in an unsupervised manner.

As features, the [discovery](#) method relies on web and corpus frequency, WordNet senses (ensuring that the adjectives have exactly two senses) and the chart parser Gsearch.

**Keller and Lapata (2003)** adopted and extended these experiments. Besides the BNC, they also sampled bigrams from the North American News Text Corpus (NANTC).

### 8.1.4. Closed Compounds

All previous approaches discussed above are focusing on [open compounds](#) (e.g., on distinguishing those from syntactic phrases and/or collocations). The [identification](#) of [closed compounds](#) can be considered as a subtask of [compound splitting](#), i.e., a [compound splitter](#) has to decide whether a single [word](#) is complex and thus subject to [decompounding](#). In the automatic compilation of the [ENCD](#) (which will be presented in Chapter 12), we proposed to use a rudimentary [compound splitting](#) method for [identifying closed compounds](#).

Considering [compound splitting](#) as an extended [closed compound identifier](#), we refer to a detailed discussion about previous work on [compound splitting](#) methods in Chapter 16.

### 8.1.5. General MWEs

Instead of focussing on **compounds**, the task of **MWE identification** (with **compounds** as a subgroup) is a predominant topic in the area of **identifying** expressions. Presenting an exhaustive list of previous work addressing the **identification** or **discovery** of **MWEs** would exceed the scope of this thesis. Instead, we present the most important and influential work. More details about the **identification** and **discovery** of **MWEs** is provided by Constant et al. (2017).

### Cross-lingual Methods

We first present related work that **identify** or **discover MWEs** with a **cross-lingual** support.

**Melamed (1997a)** developed a method for automatically **discovering** English **MWE** entities that he calls **Non-Compositional Compounds** (NCCs). However, the group of NCCs also includes kinds of **MWEs** that are not generally considered to be **compounds**, such as named entities (e.g., *Ottawa River*), idiomatic expressions (e.g., *kick the bucket*), **complex nominals** (e.g., *cry for help*) or verbal collocations (e.g., *arrange a meeting*).

The **discovery** method is unsupervised and relies on **parallel data**.

By comparing the “predictive power” between two translation models differing in the fact of whether a certain **word** sequence is treated as NCC or not, Melamed (1997a) detects non-compositional compounds. As feature of the objective function, he used **Mutual Information** (MI). For two translation models  $TM_{trial}$  (involving a NCC candidate) and  $TM_{basic}$  (without NCC), the NCC candidate is considered as valid if the value of  $TM_{trial}$ ’s objective function is higher than the value for  $TM_{basic}$ .

**Moirón and Tiedemann (2006)** used statistical **word** alignment across languages for classifying Dutch **MWEs** “along a continuum ranging from literal and transparent expressions to idiomatic and opaque expressions”. As aligned languages, Moirón and Tiedemann (2006) used English, Spanish and German, as they occur in the **parallel** Europarl corpus. The presented experiments were based on the assumption that an expression with a literal meaning has a translation which is the combination of the translations of its **constituents**, and a non-compositional expression does not have a translation which corresponds to the combination of its **constituents**’ translations.

In the first step, the Dutch portion of Europarl was parsed using Alpino<sup>3</sup>. From these parses, 191K tuples of main verb and **PP** argument were collected. Using a combination

---

<sup>3</sup><http://www.let.rug.nl/~vannoord/alp/Alpino>

## 8. Related Work on Compound Identification

of three statistical metrics (e.g., **AMs**), all **MWE** candidates were ranked and the top 200 **MWEs** were sampled.

Two **cross-lingual** measures were used for re-ranking the **MWE** candidates. Therefore, all observed translations in the **parallel corpus** are collected.

Firstly, assuming that it is harder to perform a consistent **word** alignment for opaque **MWEs**, Moirón and Tiedemann (2006) calculated the average translation entropy (Melamed, 1997b), where idiomatic expressions were expected to have a higher entropy.

Secondly, the proportion of default translations (derived from an automatically compiled link lexicon of all **word** alignments) among all observed translations for an **MWE** was used as second ranking measure. Moirón and Tiedemann (2006) observed significant improvements in the ranking over the **AM** baseline for both **cross-lingual** measures.

**Caseli et al. (2009)** presented a statistical and a **word** alignment-based method for automatically **discovering** Portuguese out-of-context **MWEs**. Although Caseli et al. (2009) conducted experiments on the Portuguese-English language pair, their method can be easily adapted to other languages (e.g., by modifying the employed **PoS patterns**).

For their **identification** method, Caseli et al. (2009) used the **parallel** Pediatrics corpus including 283 Portuguese texts.

From this corpus, the Pediatrics Glossary, a gold standard for evaluating the presented **discovery** methods, was semi-automatically compiled: *N*grams with a minimum frequency of 5 were extracted and cleaned using a **PoS pattern** (e.g., truncating leading determiners). These *N*grams were manually checked by human annotators, leading to a set of 2407 **terms** (bigrams, trigrams and a few *M*grams ( $M > 3$ )).

In the statistical approach, Caseli et al. (2009) extracted 65K bigrams and 55K trigrams having a minimum frequency of 2 from the Portuguese portion of the Pediatrics corpus. They ranked the candidate **MWEs** according to various **AMs**, such as **MI**, **PMI**,  $\chi^2$  or log-likelihood.

In contrast to our **identification** method which will be presented in Chapter 11, Caseli et al. (2009) did not restrict to **closed compounds** in their **word** alignment based approach. “The method looks for the sequences of source **words** that are frequently joined together during the alignment despite the number of target **words** involved”.

In the first step, source **MWE** candidates are extracted if “they are linked to the same target unit”. That is, if a (Portuguese) source multi-**word** sequence  $S = s_1 \dots s_n$  (with  $n \geq 2$ ) is aligned to any (English) target **word** sequence  $T = t_1 \dots t_m$  (with  $m \geq 1$ ),  $S$  is considered as a possible **MWE**, such as the Portuguese *aleitamento materno* being aligned to the English *breastfeeding*. Restricting to an alignment  $n : m$  ( $n \geq 2$ ), the list of



**MWE** candidates can be considered as a refinement of the phrase tables in phrase-based **SMT**.

In the next step, the **MWE** candidates are filtered using some predefined **PoS patterns** and a frequency threshold. Caseli et al. (2009) discussed three **PoS patterns** and frequency thresholds: (1) the same as for building the Pediatrics Glossary, (2) excluding **MWEs** that match “patterns beginning with determiner, auxiliary verb, pronoun, adverb, conjunction and surface forms such as those of the verb *to be*, relatives (*that, what, when, which, who, why*) and prepositions (*from, to, of*)” or occurring less than 2 times, and (3) excluding **MWEs** that match “patterns beginning or finishing with determiner, adverb, conjunction, preposition, verb, pronoun and numeral” or occurring less than 2 times.

**Caseli et al. (2010)** adopted the **word** alignment approach of Caseli et al. (2009) for **discovering MWEs** of both languages’ sides in the preprocessed **parallel** (Portuguese-English) Brazilian scientific magazine *Pesquisa FAPESP*<sup>4</sup> corpus.

**Ramisch et al. (2010a)** developed a hybrid method for the **discovery** of English and Portuguese **MWEs** that is based on a combination of statistical information (in terms of **AMs**) and **word** alignment, two **discovery** features which have been compared by Caseli et al. (2009).

As gold standard, they compiled the Pediatrics Glossary for both English and Portuguese.

The list of **MWE** candidates of both individual approaches were filtered using some rules:

- punctuation, numbers and special characters (such as dashes, brackets, etc. . . ) are removed from all **MWE** candidates
- **MWE** candidates having a frequency below 5 are removed
- following the **PoS patterns** used by Caseli et al. (2009), **MWE** candidates starting with function words (determiners, auxiliaries, pronouns, adverbs, conjunctions, forms of the verb *to be* and prepositions) are removed

A main difference between the statistical approach and the alignment-based approach is that the latter is able to capture discontinuous **MWEs**, whereas the statistical approach is designed for expressions containing contiguous **words**.

---

<sup>4</sup><http://www.revistapesquisa.fapesp.br>

Ramisch et al. (2010a) combined both [discovery](#) approaches using a Bayesian network classifier. As input, they used the filtered list of [MWE](#) candidates from the statistical approach, annotated with all used [AMs](#) and the boolean judgement of [MWE](#)-hood from the alignment-based approach.

In [Ziering et al. \(2013b\)](#), we tried to mitigate the negative impact of semantic drift (i.e., gradual degradation) on [semantic lexicon bootstrapping](#) by using a [multilingual ensemble method](#), where the processes of several [lexicon bootstrappers](#) in different languages are iteratively combined and the consensus [terms](#) (i.e., both single nouns and nominal [MWEs](#)) are retained.

For this ensemble method, we developed an automatic [multilingual term](#) extractor based on a preprocessed [parallel corpus](#). First, we defined a *term-specifying language*  $L_{term}$  (i.e., the language that specifies the set of candidate [terms](#) for all languages). The language  $L_{term}$  should be a [closed compounding language](#) such as German. Furthermore, German is an ideal candidate for  $L_{term}$ , because nouns are capitalized and thus, there is no need for [PoS](#) information. German [terms](#) are defined as a capitalized token with at least four letters. For each unordered language pair in the [parallel](#) setup,  $\{L_{term}, L_i\}$ , a [term](#) in  $L_i$  is defined as a [word](#) sequence that is aligned to a [term](#) in  $L_{term}$ .

For example, the English [word](#) sequence *liquid phase hydrogenation* is classified as [term](#), because it is aligned to the German [nominal compound](#) *Flüssigphasenhydrierung*.

For optionally suppressing noise due to [word](#) alignment errors, we applied [PoS](#) taggers on the aligned languages and filtered aligned [term](#) candidates that do not conform with a [PoS pattern](#) (shown as regular expression in [Formula 8.2](#)), which was adapted from Justeson and Katz (1995).

$$(\text{ADJ}|\text{GERUND}|\text{NOUN})^* \text{NOUN} (\text{PREP NOUN}^+)? \quad (8.2)$$

## Monolingual Methods

Next, we present previous work on the monolingual [MWE identification](#) and [discovery](#).

[Ramisch et al. \(2010c\)](#) developed the `mwetoolkit`, a language-independent rule-based extraction method for [MWEs](#). While all experiments are conducted for English [MWEs](#), it is plausible that this method could be easily adapted to other languages.

The `mwetoolkit` is an automatic and unsupervised approach to extracting out-of-context [MWE](#) types.

The method is divided into two steps: candidate generation and candidate filter. In the generation step, possible [MWE](#) candidates are extracted from a monolingual

preprocessed corpus using shallow linguistic features such as the [word forms](#), [lemmas](#), [PoS patterns](#) and combinations thereof (e.g., ‘*take* NOUN’). In the second step, a filter based on [AMs](#) (such as maximum likelihood estimator, Dice’s coefficient, [Pointwise Mutual Information](#) (PMI) or Student’s t-score) is applied to all candidates and those having a value above a predefined threshold are retained.

In [Ziering et al. \(2013a\)](#), we developed a technique for [bootstrapping](#) a semantic lexicon for English nominal [terms](#) (including both single nouns and [MWEs](#)) using coordination patterns. The motivation for using coordinations as grounding for [semantic lexicon bootstrapping](#) is that they are likely to contain co-hyponyms, i.e., instances of the same semantic class (e.g., substances as in ‘silver and gold’).

For automatically sampling out-of-context [terms](#) being subject to semantic classification, we used a [PoS pattern](#) as regular expression:  $(\text{ADJ}|\text{NOUN})^* \text{NOUN}$ , i.e., any sequence of adjectives or nouns followed by the [head](#) noun. This [term](#) pattern is integrated into two [PoS patterns](#) modelling coordinations: [and/or](#) coordinations (Formula 8.3) and [punctuation](#) coordinations (Formula 8.4).

$$((\text{TERM}, )^*|(\text{TERM}; )^*) \text{TERM} ((\text{and}(/or)?|or) \text{TERM})^+ \quad (8.3)$$

$$((\text{TERM}, )^+ \text{TERM}, \text{TERM})|((\text{TERM}; )^+ \text{TERM}; \text{TERM}) \quad (8.4)$$

An [and/or](#) coordination consists of two parts: (1) a list in which [terms](#) are separated by commas or semicolons and (2) [terms](#) separated by *and*, *or* or *and/or*. While the first part can be empty, the second part has to contain at least two [terms](#), i.e., we did not interpret a single [term](#) as a unary coordination.

A [punctuation](#) coordination can be considered as the first part of an [and/or](#) coordination with a minimum size of two [terms](#).

The usage of coordination patterns provides some benefits for the [discovery](#) of [terms](#): coordinated [NPs](#) tend to be less modified, less complex and the context of an [NP](#) within the coordination makes it easier to determine its boundaries; the internal boundaries are always connectors (i.e., commas, semicolons or conjunctions).

## 8.2. Compound Resources

In this section, we present the most relevant previous related work that addresses the compilation and content or usage of [compound](#) resources. For the sake of simplicity,

we do not discuss difficulties of specific annotations (e.g., of the annotation of [semantic relations](#)), because this would exceed the scope of this thesis. Instead, we focus on **seven resource features**, which are similar to the methodological features described in Section 8.1:

1. **Compound class** - Is the resource limited to a certain spelling form (e.g., [closed compounds](#) or [open compounds](#)) or to a certain PoS category of [compounds](#) (e.g., [nominal compounds](#), [noun compounds](#) or [verbal compounds](#))? In this section, we will also broach the area of general [MWEs](#) including [compounds](#) or other types of [MWEs](#).

The [ENCD](#) and the [ENCR](#) gold standard restrict to the most frequent [head](#) category, viz. the nominal [head](#), and thus only contains [nominal compounds](#). In contrast to many other resources, both the [ENCD](#) and the [ENCR](#) comprise [closed](#), [hyphenated](#) and [open compounds](#).

2. **Language** - Which language(s) does the resource provide [compounds](#) for?

The [target language](#) of the [ENCD](#) is English, but our database also provides corresponding semantic equivalents in various aligned European languages.

The [identified nominal compounds](#) in the [ENCR](#) gold standard are in English.

3. **Compilation** - What manual or automatic [identification/discovery](#) methods or heuristics have been used for building the resource?

The [ENCD](#) has been compiled automatically with a [cross-lingual identification](#) method which will be presented in Chapter 11.

The [ENCR](#) gold standard has been annotated manually by two trained linguists, as will be described in Chapter 13.

4. **Contextuality** - Is the resource a [compound](#)-annotated corpus or a lexicon which lists [compound](#) types out-of-context?

The [ENCD](#) provides [nominal compounds](#) in the monolingual and [cross-lingual](#) context of one sentence (i.e., the English sentence surrounding the [identified nominal compound](#) and the aligned sentences in up to nine European languages).

The [ENCR](#) gold standard also provides [nominal compounds](#) in the monolingual context of one sentence. In both resources, we point to the position of the sentence in the corresponding Europarl document, such that a reconstruction of a larger surrounding context (both monolingual and [cross-lingual](#)) is possible.

5. **Size** - How many **compound** tokens or types have been collected?

As shown in Table 12.1, the size of the **ENCD** ranges between 137K and 2M **compound** tokens (depending on the degree of **cross-lingual closed compounding**,  $\Xi_{closed}$ ).

The **ENCR** gold standard (**Combination**) comprise 824 **compound** tokens in 394 sentences.

6. **Purpose** - Is the resource theoretically motivated (e.g., aimed to serve for linguistic research on **compounding**) or practically motivated (e.g., as a gold standard for a **compound** analysis task such as **compound splitting**)? These different purposes result in different additional information, where the practically oriented datasets usually restrict to the information which is necessary for the **NLP** task at hand. The **ENCD** and the **ENCR** gold standard are located on the theoretically motivated end of the scale but also contain various types of information serving for different tasks of **compound** analysis.

7. **Additional information** - Which information is provided along with the **identified/discovered compounds**? For example, resources functioning as **compound splitting** gold standards provide the **split point** and/or the composed **lemmas**.

In the **ENCD**, we provide morphological information, the **lemmas**, the **word** alignments between the languages and some further formats (as discussed in Section 12.2).

In the **ENCR** gold standard, we provide human ratings for the **compoundhood** and for six properties (e.g., prosody) which are mentioned in the linguistic literature as criteria for the **compoundhood** definition (as discussed in Chapter 4).

For structuring this section, we group previous work with respect to the compound class feature.

### 8.2.1. General Compounds

The most similar work to the **ENCD** is the “**multilingual** database of **compound** words” developed by **Guevara et al. (2006)**, which is resulted from the **MORBO/COMP (Morphology at Bologna University - compounds)** project<sup>5</sup>, which aimed to collect **compound** information “in a standardized manner” allowing for **cross-linguistic** comparisons. This database (also called **MORBO/COMP**) contains a wide variety of **compound** classes (e.g.,

---

<sup>5</sup>[morbocomp.sslmit.unibo.it](http://morbocomp.sslmit.unibo.it)

## 8. Related Work on Compound Identification

nominal compounds, verbal compounds, adjectival compounds, closed compounds, ...) in over 20 languages.

Basque	Bulgarian	Byelorussian	Catalan	Chinese
Dutch ✓	English ✓	Finnish	French ✓	German ✓
Greek ✓	Hebrew	Hungarian	Italian ✓	Japanese
Korean	Latin	Norwegian	Polish	Portuguese ✓
Russian	Serbo-croatian	Spanish ✓	Swedish ✓	Turkish

Table 8.1.: Languages in MORBO/COMP

The languages included in MORBO/COMP are given in Table 8.1. From the 10 languages of the ENCD, 9 are available in MORBO/COMP (Danish is missing). The common languages are marked with ✓. However, in contrast to the ENCD, the languages’ parts in MORBO/COMP are not parallel, i.e., the languages’ parts “differ in granularity and coverage”.

The resource is manually compiled by a group of native-speaking morphologists from various European languages. The *compound* classification proposed by Bisetto and Scalise (2005) (discussed in Section 3.7.1) has been used as annotation guidelines for members of the MORBO/COMP project.

The resulting resource is a *compound* lexicon, i.e., the *discovered compounds* are out of context.

Unfortunately, the MORBO/COMP database cannot be procured from any sources. Therefore, there is no information available about the size of this resource and a detailed comparison to the ENCD is not possible.

One of the ultimate goals of MORBO/COMP is to provide the first resource for typological research on compounding, i.e., to compare *compound* data across the “world’s languages” (e.g., the degree of *endocentricity/exocentricity*). Moreover, MORBO/COMP is aimed to serve for automatic *compound identification* and classification in large corpora.

Guevara et al. (2006) lists the information fields that the database provides, as shown in Table 8.2.

Some of these information fields are also available in the ENCD, e.g., the language, the *compound*, its structural description or the morpho-syntactic gender of the *constituents*.

Information field in MORBO/COMP	Available in the ENCD?
Language	✓
Compound	✓
Word category: N, V, A, P, Adv, etc.	only nominal compounds
Structural description (e.g., [V+N])	✓
Classification into 3 major classes (see Section 3.7.1)	✗
Endocentricity	✗
Position of the categorial/syntactic head	✓
Position of the semantic head	✗
Constituents	(✓)
Linking elements	✗
Morphosyntactic marking (i.e., word inflection)	✗
Gender of the constituents	✓
English gloss of compound and constituents	✗

Table 8.2.: Information fields in MORBO/COMP

### 8.2.2. Two-Noun Compounds

The resource of **Rosario and Hearst (2001)** contains English 2NCs from the medical domain. Rosario and Hearst (2001) automatically compiled this resource from search results in MedLine, containing “references and abstracts from 4300 biomedical journals”. Therefore, they used several search queries for covering various medical topics. From the search results, titles and abstracts were extracted, and the data was preprocessed (e.g., PoS-tagged). Sequences of exactly two nouns were identified. In the next step, Rosario and Hearst (2001) used the Unified Medical Language System (UMLS) (Humphreys et al., 1998) and retained only those 2NCs whose constituents can be mapped onto a corresponding term in the medical ontology MeSH (Lowe and Barnett, 1994).

This resource provides 2NC types out of context.

The final dataset contains 2245 English 2NCs.

Rosario and Hearst (2001) developed a classification algorithm for the identification of the semantic relation holding between modifier and head of 2NCs. This algorithm is evaluated using the compiled dataset.

Rosario and Hearst (2001) defined 38 relations which form the additional annotation of the identified 2NCs. For example, PHYSICAL PROPERTY as in *blood pressure*, FREQUENCY as in *headache interval* or MISUSE as in *acetaminophen overdose*.

The resource of **Kim and Baldwin (2005)** contains English 2NCs annotated with

## 8. Related Work on Compound Identification

semantic relations. Kim and Baldwin (2005) extracted **2NCs** from the Wall Street Journal (WSJ) component of the Penn Treebank. For their WordNet-based approach, they disregarded **2NCs** including any proper nouns. If a **2NC** is part of a larger **noun compound** (e.g., of a **3NC**), it is also excluded from the further annotation.

The final set contains 2169 out-of-context **2NCs**.

Kim and Baldwin (2005) proposed a method for recognizing the semantic relationship between the **constituents** of novel **binary noun compounds** using WordNet Similarity. The compiled dataset is used as training set (1088 samples) and as test set (1081 samples).

The **2NCs** are annotated with semantic relations from an inventory of 20 relations, e.g., TOPIC as in *computer expert*, PURPOSE as in *concert hall* or OBJECT as in *horse doctor*.

The dataset of Ó Séaghdha (2007) contains English **2NCs** annotated with a **semantic relation** class. Ó Séaghdha (2007) compiled this resource using an heuristics which is similar to the approach of Lapata and Lascarides (2003): after preprocessing BNC, all **2NCs** which are not adjacent to another noun and consist of only alphabetic characters are extracted. From the resulting set, Ó Séaghdha (2007) randomly selected 2000 compound type samples that were subsequently annotated with the semantic relation class. While Lapata and Lascarides (2003) reported a **discovery** accuracy of 70.3%, the restriction to **2NCs** consisting of only alphabetic characters yield an accuracy of 78.4%.

The 2000 **noun compounds** in this resource are extracted out of context.

Ó Séaghdha (2007) described the development of a new annotation scheme for semantic relation classes of **2NCs**. He applied this annotation scheme on the compiled

Relation class	Frequency
BE	191 (9.6%)
HAVE	199 (10%)
IN	308 (15.4%)
ACTOR	236 (11.8%)
INST	266 (13.3%)
ABOUT	243 (12.2%)
Total	1443 (100%)

Table 8.3.: Semantic relation class frequency distribution in Ó Séaghdha (2007)

dataset, resulting in 1443 **noun compounds** annotated with one out of six coarse semantic relation classes: BE (e.g., a **substance\_form** as in *plastic box*), HAVE (e.g., a



## 8. Related Work on Compound Identification

`part_whole` relation as in *car door*), `IN` (e.g., a `spatially_located_event` as in *dining room*), `ACTOR` (e.g., a `participant_event` relation as in *student demonstration*), `INST` (e.g., the `non_sentient_participant_event` relation as in *machine translation*) and `ABOUT` (e.g., a `topic_object` relation as in *fairy tale*). The distribution of these classes is given in Table 8.3.

The dataset of **Girju (2007)** contains `2NCs` and `complex nominals`, consisting of two nouns and a preposition in between, in *English* (`target language`) and in five parallel Romance languages: *Spanish, Italian, French, Portuguese* and *Romanian*. Girju (2007) semi-automatically compiled this dataset and used as source two `parallel corpora`: Europarl and CLUVI<sup>6</sup>. In the Europarl corpus, for Spanish, Italian, French and Portuguese, a bitext with English is created and `word` alignment is performed. Then, the English sentences which are common for all four bitexts are considered. This set of sentences is syntactically parsed. In CLUVI, Girju (2007) focused on English paired with Portuguese and Spanish. Using the CLUVI search interface, a set of 2800 aligned sentences was created. After parsing the English parts, `noun-noun` and `noun-preposition-noun` sequences were manually aligned to their translations. In order to complement the set of equivalents for both corpora, native speakers for Romanian, Italian and French provided the correct translations from the English samples.

From Europarl, the first 3000 instances of `NPs` were extracted out of context, where 48.8% were `2NCs` and 51.2% were `complex nominals`. For CLUVI, the final set contains 2200 English `NPs` distributed in 26.77% `2NCs` and 73.23% `complex nominals`.

The goal of Girju (2007) was to classify semantic relations in `NPs`, (i.e., `2NCs` or `complex nominals`) using `cross-lingual` evidence from `parallel data` including the English `target language` and five aligned Romance languages. The key feature of aligned Romance `complex nominals` is the preposition which correlates with the semantic interpretation, as discussed in Section 5.2. For the semantics annotation, nominal `constituents` were mapped onto a WordNet sense. If one `constituent` was unknown to WordNet, the `compound` is removed. Then, each sample is annotated with a semantic relation, leading to the final dataset of 2954 (Europarl) and 2169 (CLUVI) samples with the format  $\langle \text{NP}_{EN}, \text{NP}_{ES}, \text{NP}_{IT}, \text{NP}_{FR}, \text{NP}_{PT}, \text{NP}_{RO}, \text{semantic relation} \rangle$ , for example,  $\langle \textit{development cooperation}; \textit{cooperaci3n para el desarrollo}; \textit{cooperazione allo sviluppo}; \textit{coop3ration au d3veloppement}; \textit{cooperare pentru dezvoltare}; \text{PURPOSE/FOR} \rangle$ .

The **BNC Compound Nominalization Dataset**, created by **Nicholson and**

---

<sup>6</sup>CLUVI - Linguistic Corpus of the University of Vigo - Parallel Corpus 2.1 - <http://sli.uvigo.es/CLUVI/>

**Baldwin (2008)**, contains English 2NCs and focuses on a deeper analysis for the subgroup of **compound** nominalizations: derivational origin of the **head** and the semantic relation holding between **modifier** and **head**. Nicholson and Baldwin (2008) manually compiled this resource from a random sample of 1000 sentences of the British National Corpus (BNC: Burnard (2000)). The **identification** and annotation of the **compounds** have been performed within the annotation process of the 1000 sentences. For this, three non-specialist annotators were employed. In the case of disagreements, an adjudicator decided and finally formed the gold standard. The annotation guidelines include four annotation steps:

1. **identify binary noun compounds** (i.e., sequences of two nouns that function as a single noun)
2. **noun compounds** that include any proper nouns are labelled as such and excluded from further annotation (PN)
3. for **noun compounds** that have a deverbal head and whose meaning relates to the verbal meaning, the underlying verb is determined.

Three categories are available:

- the **head** is not deverbal (NV)
- the **head** is deverbal, but “it does not occur in a productive semantic relation with the **modifier**” (NA)
- the **head** is deverbal and “forms the basis of a semantic relation with the **modifier**” (SUB, DOB or POB)

4. the implicit semantic relation between **modifier** and **head** of **compound** nominalizations is determined.

Three semantic relations are available:

- the **modifier** corresponds to the **subject** of the underlying verb (SUB)
- the **modifier** corresponds to the **direct object** of the underlying verb (DOB)
- the **modifier** corresponds to a **prepositional object** of the underlying verb (POB)

In this case, the appropriate preposition has to be provided.

As an annotation example, Nicholson and Baldwin (2008) refer to the sentence

## 8. Related Work on Compound Identification

*Vibration to the platform caused the power supply to be disrupted when the generators stopped, creating a temporary disruption to production and affecting the drilling operation.*

First, the annotators **identify** *power supply* and *drilling operation*. In the next step, both **noun compounds** are retained as containing no proper nouns. As underlying verb, they determine *supply* and *operate*, respectively. Finally, the implicit **semantic relation** for both compound nominalizations is the direct-object relation. With respect to the gold standard, the three annotators had an average precision of 92.5% and an average recall of 84.8%, while they had an inter-annotator agreement rate of 98.4% with a Kappa coefficient (Carletta, 1996) of  $\kappa = 0.83$ , indicating good agreement.

The BNC Compound Nominalization Dataset provides **noun compounds** in context.

As stated by Nicholson and Baldwin (2008), 32% of the sentences (i.e., 320 sentences) contain one or more **noun compounds**. In total, 464 **noun compounds** were annotated, where 119 **noun compounds** include proper nouns (PN). For the remaining 345 **noun compounds**, Table 8.4 shows the distribution of the five possible verbal classes.

Verbal class	Example	Frequency
subject (SUB)	<i>eyewitness report</i>	22 (6.4%)
direct object (DOB)	<i>eye irritation</i>	63 (18.2%)
prepositional object (POB)	<i>side show</i>	44 (12.8%)
not deverbal (NV)	<i>scout hut</i>	58 (16.8%)
no productive semantic relation (NA)	<i>memory size</i>	158 (45.8%)

Table 8.4.: Distribution of verbal classes in Nicholson and Baldwin (2008)

The goal of the **BNC Compound Nominalization Dataset** is to serve for training and testing models on the **identification** and interpretation of **compound** nominalisations.

One example of a **compound**-annotated sentence is given in Figure 8.1.

```
<doc>
Demand for the new car is strongest in large urban areas like New
<cn rel="PN" hvf="">York city</cn>, Los Angeles and Miami , where
bombings , riots and car-jackings fill the <cn rel="NA"
hvf="bulletin">news bulletins</cn> .
</doc>
```

Figure 8.1.: Sample data in Nicholson and Baldwin (2008)

## 8. Related Work on Compound Identification

The resource of **Tratz and Hovy (2010)** comprises English **2NCs**. Tratz and Hovy (2010) manually compiled a dataset from two sources: (1) “an in-house collection of terms extracted from a large corpus using part-of-speech tagging and mutual information” and (2) the WSJ component of the Penn Treebank. **Noun compounds** that include or constitute proper nouns were disregarded.

Tratz and Hovy (2010) claim that this dataset is “the largest **noun compound** dataset yet produced” and contains 17,509 out-of-context **2NC** types.

Tratz and Hovy (2010) addressed the task of determining the implicit semantic relation of **binary noun compounds**. In contrast to previous work, they presented a much larger taxonomy of 43 relations. The authors developed a supervised classifier which is trained on the compiled dataset annotated with the 43 semantic relations.

The semantic relation inventory includes relations such as POSSESSOR + OWNED/POSSESSED as in *family estate* or PERFORM/ENGAGE\_IN as in *cooking pot*.

The dataset of **Reddy et al. (2011)** contains English **2NCs** annotated with human compositionality ratings. Reddy et al. (2011) automatically collected **2NCs** from WordNet with varying literalness of **modifier** and **head**, leading to four classes (i.e., literal/non-literal **modifier/head**). They used some WordNet-based heuristics for determining literalness (i.e., **2NCs** whose **constituents** are hypernyms of the **2NC** or occur in its definition). In the next step, for each class, 30 samples were randomly selected. In the final step, samples having a minimum frequency of 50 in the ukWaC corpus are retained.

The final dataset consists of 90 out-of-context **noun compounds**.

The work of Reddy et al. (2011) deal with the rating of compositionality of **noun compounds** using Amazon Mechanical Turk<sup>7</sup> (AMT). The **discovered compounds** are associated with compositionality ratings for **modifier**, **head** and the whole **compound**, including the standard deviation, as exemplified in Table 8.5.

<b>2NC</b>	<b>Modifier</b>	<b>Head</b>	<b>Compound</b>
<i>climate change</i>	4.90±0.30	4.83±0.38	4.97±0.18
<i>ivory tower</i>	0.38±1.03	0.54±0.68	0.46±0.68
<i>video game</i>	4.50±0.72	5.00±0.00	4.60±0.61
<i>diamond wedding</i>	1.07±1.29	3.41±1.34	1.70±1.05

Table 8.5.: Examples of compositionality ratings by Reddy et al. (2011)

The dataset of **Graves et al. (2013)** contains English **2NCs** and accompanying hu-

---

<sup>7</sup>mturk.com

## 8. Related Work on Compound Identification

man ratings about meaningfulness. Graves et al. (2013) extracted this dataset from existing resources. A key criterion for the selection of 2NCs was that “all nouns making up these phrases were rated as relatively high in imageability, a dimension closely related to concreteness”, whereas noun compounds including abstract constituents were excluded. For compiling the dataset, Graves et al. (2013) automatically collected the 500 most concrete words from a mixed database for imageability derived from previous imageability rating studies. Using the CELEX lexical database (Baayen et al., 1995), words are removed if their PoS probabilities do not point to a nominal category. For the remaining set of nouns, all pairwise combinations are generated. In the next step, a database of human-generated text from Usenet groups (Shaoul and Westbury, 2007) is used for finding corpus evidence. Noun pairs having evidence in only one direction are retained.

Using a Web interface, the noun pairs were annotated with meaningfulness ratings by 150 psychology students.

The resulting resource is a compound lexicon which provides 2NCs out of context.

The final dataset contains 2,160 English binary noun compounds, where one half (containing 1080 samples) is the inversion of the other half (e.g., *ski jacket* and *jacket ski*).

Graves et al. (2013) researched the *conceptual combination*<sup>8</sup> forming 2NCs. The motivation for the authors’ study is that 2NCs “once rated in forward (meaningful) and reversed orders, could be used to examine various aspects of combinatorial processing”.

Compound	Meaningfulness	Inverse	Meaningfulness
<i>school pony</i>	2.5	<i>pony school</i>	4
<i>tomato juice</i>	4	<i>juice tomato</i>	4
<i>prison door</i>	4	<i>door prison</i>	2
<i>child seat</i>	4	<i>seat child</i>	1
<i>wine bottle</i>	4	<i>bottle wine</i>	1.5

Table 8.6.: Meaningfulness ratings in Graves et al. (2013)

The dataset of Farahmand et al. (2015) contains English 2NCs annotated with boolean features for compositionality and conventionalization. A compound is conventionalized if its constituents cannot be replaced by a near-synonym “according to some cultural or historical convention” (Farahmand et al., 2015). Farahmand et al.

<sup>8</sup>the “combining of a pair of words, each representing a distinct concept, into a phrase that derives its meaning from both words” Graves et al. (2013)

## 8. Related Work on Compound Identification

(2015) semi-automatically extracted this resource from a cleaned version of the English Wikipedia. The most frequent sequences of two nouns were extracted (leading to a set of 169,000 [word](#) pairs). This set of [2NCs](#) is stratified into five groups according to their bigram frequency (ensuring that each group has approximately the same number of [2NCs](#)). From each group, 250 [noun compounds](#) are randomly extracted. For mitigating the impact of the correlation between compositionality and frequency, two experts add 100 partly and fully non-compositional [2NCs](#) to the set. Finally, the dataset was manually filtered (e.g., in cases of [PoS](#) errors).

<b>Non-compositional</b>	<b>Compositional but conventionalized</b>	<b>compositional and not conventionalized</b>
<i>battle cry</i>	<i>bulletin board</i>	<i>area director</i>
<i>flag stop</i>	<i>cable car</i>	<i>art collection</i>
<i>gun dog</i>	<i>car chase</i>	<i>ankle injury</i>
<i>jet lag</i>	<i>food court</i>	<i>animal life</i>
<i>lead time</i>	<i>wish list</i>	<i>bus service</i>
<i>face value</i>	<i>speed limit</i>	<i>computer usage</i>
<i>mind map</i>	<i>background check</i>	<i>wrestling fan</i>

Table 8.7.: Examples of 2NCs in the dataset of Farahmand et al. (2015)

The final resource contains 1048 out-of-contexted [2NC](#) types.

As claimed by Farahmand et al. (2015), this dataset is intended to serve for the evaluation of methods for [MWE identification](#).

Some examples presented by Farahmand et al. (2015) are shown in Table 8.7.

### 8.2.3. Binary Nominal Compounds

The dataset of [Nastase and Szpakowicz \(2003\)](#) contains English [binary nominal compounds](#) having a nominal [head](#) and a [modifier](#) which is either a noun, an adjective or an adverb.

For their experiments about the similarity between base [NPs](#), Nastase and Szpakowicz (2003) manually collected samples from the data of Levi (1978), automatically from Larrick (1961) and semi-automatically from *SemCor* (which is annotated with WordNet 1.6 senses).

The final resource comprise 600 out-of-context base [NPs](#).

Nastase and Szpakowicz (2003) addressed the task of exploring the semantic similarity of [compounds](#) (or noun-[modifier](#) constructions) clustered according to their semantic

relations between [modifier](#) and [head](#).

All samples are manually annotated with semantic relations from an inventory of 30 relations, e.g., CAUSE as in *flu virus* or EFFECT as in *exam anxiety*.

### 8.2.4. Closed Compounds

For [closed compounding languages](#) (e.g., German), there are datasets serving as gold standards for the task of [compound splitting](#) (i.e., for training or testing a [splitting](#) method). Usually, these datasets contain the [closed compound](#) and its [constituents](#). In some cases, not all [constituents](#) are presented but only the two [immediate constituents](#) (e.g., the GermaNet [closed compound](#) dataset by Henrich and Hinrichs (2011)) and in other cases, the [constituents](#) are represented as [constituent forms](#) (e.g., in the gold standard of Holz and Biemann (2008)) or as [constituent lemmas](#) (e.g., in the data of Henrich and Hinrichs (2011)). Finally, some gold standards only provide the [constituents](#), whereas others provide information about the concrete morphological transformations the [constituents](#) have undergone (e.g., the [constituent inflection](#)) (e.g., in the gold standard of Marek (2006)).

We will address the task of [multilingual compound splitting](#) in Part D. The gold standards used in our [splitting](#) experiments will be presented in Section 18.6.4. Further gold standards for [compound splitting](#) will be discussed in Appendix D.

In the ENCD, for [closed compounds](#) we provide the [lemmas](#) of the two [immediate constituents](#), derived from the rudimentary binary [compound splitter](#) (9.2).

### 8.2.5. General MWEs

The Wiki50 corpus, developed by Vincze et al. (2011), contains English instances of [MWEs](#) in general including [open nominal compounds](#) and [adjectival compounds](#). Vincze et al. (2011) manually compiled the Wiki50 corpus from a randomly selected set of 50 Wikipedia articles with a running text of at least 1000 words. For each instance of [MWE](#), the class is annotated (e.g., light-verb constructions or named entities). For [compounds](#), Vincze et al. (2011) restricts to [nominal compounds](#) and [adjectival compounds](#), arguing that these are the only productive [compound](#) categories. For the annotation process, two linguists were employed. From the set of 50 Wikipedia articles, 15 were annotated by both linguists for measuring [Inter-Annotator Agreement \(IAA\)](#). As stated by Vincze et al. (2011), for [nominal compounds](#), there was an agreement of about 71% for precision, recall and  $F_1$ , a Jaccard coefficient of 0.5518 and a Kappa score of  $\kappa = 0.6414$ . A major

## 8. Related Work on Compound Identification

reason for the disagreement were embedded **MWEs**. While the annotation instructions required to mark the longest units, the annotators happen to only **identify** a part of the **MWE**. As future work, Vincze et al. (2011) plan to also annotate embedded **MWEs**.

<b>MWE type</b>	<b>Token count</b>	<b>Type count</b>
Named Entity - <b>Person</b>	4087 (31.9%)	1050 (21.9%)
<b>Nominal compound</b>	<b>2926 (22.8%)</b>	<b>1371 (28.6%)</b>
Named Entity - <b>Misc</b>	1819 (14.2%)	716 (14.9%)
Named Entity - <b>Location</b>	1557 (12.1%)	599 (12.5%)
Named Entity - <b>Organization</b>	1496 (11.7%)	625 (13.0%)
Verb-particle construction	446 (3.5%)	210 (4.4%)
Light-verb construction	368 (2.9%)	138 (2.9%)
<b>Adjectival compound</b>	<b>78 (0.6%)</b>	<b>51 (1.1%)</b>
Other <b>MWE</b> type	21 (0.2%)	17 (0.4%)
Idiom	19 (0.1%)	17 (0.4%)
Total	12,817	4794

Table 8.8.: MWE distribution in the Wiki50 corpus

This resource is a **MWE**-annotated corpus and provides the context for each **identified MWE**.

In total, there are 12,817 annotated **MWEs**. Table 8.8 shows the distribution of the various **MWE** types in the Wiki50 corpus. While there are more named entity tokens of the type **Person** than **nominal compounds**, considering the types, the **nominal compounds** are the majority class of **MWEs** in Wiki50.

This resource is aimed to facilitate the **identification** of **MWEs** and can be used for training and testing methods for **MWE identification** and **Named Entity Recognition (NER)**.

An example of a **MWE**-annotated sentence (with the BIO tagset) is given in Figure 8.2.

The Comprehensive Multiword Expressions (CMWE) Corpus, created by **Schneider et al. (2014b)**, contains English **MWEs** of various kinds (including **open** and “conventionally” **closed compounds**). However, Schneider et al. (2014b) did not indicate the kind of **MWE** (e.g., **noun compound** vs. light-verb construction) which has been done for the Wiki50 corpus by Vincze et al. (2011).

Schneider et al. (2014b) manually enriched the REVIEW section (55,579 **words** in 3812 sentences in 723 review documents) of the English Web Treebank (Bies et al., 2012) with comprehensive **MWE** annotations. The key principles of their annotation scheme are *heterogeneity* (no restriction to a certain type of **MWE**), *shallow but gappy grouping*



## 8. Related Work on Compound Identification

```
One 0 0
of 0 0
the 0 0
oldest 0 0
methods 0 0
is 0 0
called 0 0
the 0 0
multiple B-MWE_COMPOUND_NOUN 0
tube I-MWE_COMPOUND_NOUN 0
method I-MWE_COMPOUND_NOUN B-COMPOUND_NOUN_SB
. 0 0
```

Figure 8.2.: Example sentence annotation in the Wiki50 corpus (Vincze et al., 2011)

(flat chunks, not necessarily contiguous) and *expression strength* (indication of idiomacy as either strong or weak multiword groupings). While “there are no perfect criteria for judging MWE-hood”, Schneider et al. (2014b) used some heuristics for the annotation of **MWEs**, such as semantic opacity, substitutability of **constituents** with synonyms and antonyms, the addition of modifiers, the re-arrangement of the syntactic structure or corpus frequency. The token-based annotation process allows for overlapping **MWEs** as in *threw a surprise **birthday party***, where the **noun compound** *birthday party* overlaps with the verbal construction *threw [...] birthday party*. The inter-annotator agreement ( $F_1$ , (Vilain et al., 1995)) ranges between 65% and 77% (depending on the setup).

The resource is a **MWE**-annotated corpus providing in-context **MWEs** such as **noun compounds**.

In total, the corpus includes 2378 **MWE** types. A statistics about the **PoS** distribution among the collected **MWEs** reveal that most **constituents** are nouns (2089), followed by verbs (1572) and prepositions (1002). The top frequent **PoS patterns** include noun-noun (i.e., **noun compounds**), proper noun-proper noun (i.e., named entities) and verb-preposition/particle/noun.

This resource is intended to serve as training data for methods for **MWE identification** as has been done by Schneider et al. (2014a).

Figure 8.3 shows two annotated example sentences given by Schneider et al. (2014b). Subscripts and the colored text indicate a strong coherence, whereas superscripts and underlined text indicate a weak coherence. Text which is written in boxes indicates a gap in the surrounding **MWE**.

## 8. Related Work on Compound Identification

- (1) My wife had *taken*<sub>1</sub> her '07<sub>2</sub> Ford<sub>2</sub> Fusion<sub>2</sub> *in*<sub>1</sub> for a routine *oil*<sub>3</sub> *change*<sub>3</sub>.
- (2) he was willing to *budge*<sub>1</sub> red<sub>a<sub>2</sub></sub> *little*<sub>2</sub> *on*<sub>1</sub> the price which means<sup>4</sup> a<sub>3</sub> lot<sup>4</sup> to<sup>4</sup> me<sup>4</sup>.

Figure 8.3.: Example annotations in the CMWE corpus (Schneider et al., 2014b)

### Additional MWE Resources

The project about **PAR**Sin and **M**ulti-word **E**xpressions (PARSEME)<sup>9</sup> provides a large list of **MWE** resources, some of which include **compounds**.

Savary (2000) provides the **English and French DELA dictionaries** providing out-of-context **MWEs** including many **compounds**.

**MWE** and **compound** annotations are also available in various treebanks<sup>10</sup> in various languages, for example in the Prague Dependency Treebank (Bejcek and Stranák, 2010) or the ITU-METU-Sabancı Treebank (IMST) for Turkish (Sulubacak et al., 2016).

**MWE** resources that focus on languages other than English include the **National Corpus of Polish** (Savary and Piskorski, 2011); the **Oxford Arabic Dictionary** (Arts, 2014); the **Bulgarian Sense-Annotated Corpus**<sup>11</sup>, the **Dictionary of Neologisms in Bulgarian Language**<sup>12</sup>, the **British English Source Lexicon (BESL)**<sup>13</sup>, the **Italian Syntactic-Semantic Treebank (ISST)** (Montemagni et al., 2000); the **Oxford English phonetics files**<sup>14</sup> and the **Serbian DELA e-dictionary** (Krstev, 2008).

There are also **MWE** resources that do not include **compounds**. Another type of **MWEs** for which datasets have been compiled are **multiword verbs**. Previous work that address resources for multiword verbs include Cook et al. (2008), Krenn (2008), Muischnek and Kaalep (2010) and Vincze and Csirik (2010). Besides **compounds**, one of the most prominent types of **MWEs** in **NLP** are **named entities**. Previous work that address resources for named entities include Doddingtong et al. (2004), Tjong Kim Sang (2002), Tjong Kim Sang and De Meulder (2003), Grishman and Sundheim (1995) and Chinchor (1998).

<sup>9</sup>[typo.uni-konstanz.de/parseme](http://typo.uni-konstanz.de/parseme)

<sup>10</sup>[clarino.uib.no/iness/page?page-id=MWEs\\_in\\_Parseme](http://clarino.uib.no/iness/page?page-id=MWEs_in_Parseme)

<sup>11</sup><http://metashare.ilsp.gr:8080/repository/browse/bulgarian-sense-annotated-corpus/b7d5478666cd11e281b65cf3fcb88b705fc4c009156a4a9499794778d015eaa8/>

<sup>12</sup><http://metashare.ilsp.gr:8080/repository/browse/dictionary-of-neologisms-in-bulgarian-language/7ad446f268ad11e281b65cf3fcb88b70dd4a3a216cb34a998c25fda3d4e70b2a>

<sup>13</sup><http://metashare.ilsp.gr:8080/repository/browse/british-english-source-lexicon-besl-version-22/dc410e62de6811e2b1e400259011f6eaff8112b159c346f8a910378af93ece2a>

<sup>14</sup><http://metashare.ilsp.gr:8080/repository/browse/oxford-english-phonetics-files/e986bb8ede6911e2b1e400259011f6eacf808bda74be4dc4879f8d2cf624cc4a>

## 9. Parallel Corpus

The main resource for [cross-lingual supervision](#), on which several methods and experiments in this thesis are based, is a [parallel corpus](#). “Parallel corpora - bodies of text in [parallel](#) translation, also known as bitexts — have taken on an important role in machine translation and multilingual [Natural Language Processing](#)” (Resnik and Smith, 2003). In contrast to comparable corpora (e.g., WIKIPEDIA<sup>1</sup>), the content provided in each language of a [parallel corpus](#) is taken to be **semantically equivalent**. Therefore, it is possible to determine equivalent lexical content (e.g., [noun compounds](#) and [noun phrases](#)) across languages using sentence and [word](#) alignment.

There are many types of [parallel corpora](#) varying in number of included languages, size (i.e., the total number of tokens across all languages) and domain. In this thesis, we restrict<sup>2</sup> to one of the most well-known [parallel corpora](#) for computational linguistics, EUROPARL<sup>3</sup>. The EUROPARL corpus is a resource of [parallel](#) proceeding texts from the European Parliament translated in many European languages and “has found widespread use in the [NLP](#) community”, such as for the training of an [SMT](#) system (Koehn, 2005). We are aware of the fact that the way how a [parallel corpus](#) such as EUROPARL emerges, is biased: there is a source text (e.g., the records of an English speaker in the European Parliament) that serves as basis for (possibly biased) generations in the target languages. To the best of our knowledge, we are not aware of a corpus containing unbiased [parallel](#) texts (e.g., a description of a language-independent object such as a picture) for serving as resource for the research on [cross-lingual compounding](#).

Nevertheless, we expect that [parallel corpora](#) such as EUROPARL are sufficient for getting a representative notion how [compounds](#) are translated across languages.

---

<sup>1</sup>wikipedia.org

<sup>2</sup>Nevertheless, we expect to see similar results for other [parallel corpora](#).

<sup>3</sup>statmt.org/europarl

## 9.1. Language Selection in Europarl

All [cross-lingual](#) discussions, methods and experiments in this thesis are based on the 7<sup>th</sup> release of the [parallel](#) EUROPARL corpus, comprising 21 European languages: **Romance** (*French, Italian, Spanish, Portuguese, Romanian*), **Germanic** (*English, Dutch, German, Danish, Swedish*), **Slavik** (*Bulgarian, Czech, Polish, Slovak, Slovene*), **Finnic-Ugric** (*Finnish, Hungarian, Estonian*), **Baltic** (*Latvian, Lithuanian*), and **Hellenic** (*Greek*) (Koehn, 2005).

Although the EUROPARL corpus comprises 21 European languages, the amount of common data they cover is rather small. For example, the current version of EUROPARL (version 7) provides proceedings of the European parliament ranging from 1996-2011, whereas Romanian entered the EU in 2007 (i.e., considering only proceedings for which there are Romanian equivalents would restrict EUROPARL to 5 years ( $\approx 31,3\%$ )). The more languages we use for a fully [parallel](#) representation of [compounds](#) across all inspected languages, the smaller the amount of available common data.

For getting a good trade-off between the [cross-lingual](#) coverage in EUROPARL and the exploration of different languages, we decided for nine aligned languages spread across three language families, as shown in Table 9.1, i.e., the four **Germanic closed compounding languages** *Danish, Dutch, German and Swedish*, the **Hellenic Greek**, which can be considered as a both [open](#) and [closed compounding language](#), and the four **Romance** languages *Spanish, French, Italian and Portuguese*, for which [compounding](#) is less prominent.

Germanic	Hellenic	Romance
Danish 	Greek 	Spanish 
Dutch 		French 
German 		Italian 
Swedish 		Portuguese 

Table 9.1.: EUROPARL language selection

## 9.2. Preprocessing Steps

For most methods and experiments in this thesis, the [parallel corpus](#) needs to be pre-processed. This step includes tokenization, [PoS](#) tagging and lemmatization.

While the content of different languages in [parallel corpora](#) can be considered as semantically equivalent, there are not always 1:1 correspondences for the sentences, e.g., an English sentence might be split up into two German sentences or might be part of a more complex French sentence. In order to have a representation of [parallel sentences](#) for all relevant languages, a **sentence aligner** has to be applied, which creates **pseudo-sentences**, containing one or several true sentences in the respective languages. Afterwards, a **word aligner** is applied pairwise on the pseudo-sentences between all languages. For the sake of simplicity, in this thesis we restrict to the [word alignment](#) between English and all aligned languages. This restriction is motivated by the fact that English is mostly used as [target language](#) (e.g., in [compound identification](#) or [compound parsing](#)) and an exhaustive [word alignment](#) would require to consider 45 language pairs.

In order to distinguish [atomic](#) from [closed compounds](#), there is need for a **compound splitter**. Focusing on [nominal compounds](#), we apply a [splitter](#) to each noun using a variant of the [splitting](#) method proposed by Stymne (2008), an elaborated version of the statistical approach of Koehn and Knight (2003). This method checks each noun for all possible segmentations into at most two [constituents](#) having at least two characters. All possible segmentations are scored with the geometric mean of the [constituents](#)' frequencies in EUROPARL. The highest-scored segmentation (possibly the [atomic](#) analysis) is used as [compound splitting](#) result.

### 9.3. Opus Europarl

In this thesis, we avoid applying all preprocessing steps described in Section 9.2 by ourselves. Instead, we decided to use the preprocessed EUROPARL resource of OPUS<sup>4</sup> developed by Tiedemann (2012). In OPUS the following [NLP](#) tools have been used for preprocessing EUROPARL.

For [PoS](#) tagging the English, Dutch, German, French, Italian and Spanish part of EUROPARL, **TreeTagger**<sup>5</sup> (Schmid, 1995) has been used.

For [PoS](#) tagging the Danish, Portuguese and Swedish part, the **Hunpos**<sup>6</sup> tagger has been used.

OPUS does not provide [PoS](#) and [lemma](#) information for Greek. Therefore, we used a Greek model of the **MATE**<sup>7</sup> tagger for preprocessing the Greek part of EUROPARL.

---

<sup>4</sup>[opus.lingfil.uu.se](http://opus.lingfil.uu.se)

<sup>5</sup>[cis.uni-muenchen.de/~schmid/tools/TreeTagger](http://cis.uni-muenchen.de/~schmid/tools/TreeTagger)

<sup>6</sup>[code.google.com/p/hunpos/downloads/list](http://code.google.com/p/hunpos/downloads/list)

<sup>7</sup>[code.google.com/p/mate-tools](http://code.google.com/p/mate-tools)

## 9. Parallel Corpus

The sentence alignment information provided by OPUS is restricted to language pairs rather than to the language selection described in Table 9.1. As we need sentence representations that are **parallel** in all 10 languages, we apply the OPUS sentence aligner (with English as *pivot*) to our language set and extract a total of **884,164 parallel pseudo-sentence representations**.

The **word alignment** information provided by OPUS is also based on language pairs. This means, the sentence-wise token indices have to be adapted to our updated pseudo-sentence representations. In OPUS, the **word alignment** tool **GIZA++** (Och and Ney, 2003) has been used with the symmetrisation heuristics **grow-diag-final-and** (Koehn et al., 2007).

# 10. Pilot Studies on Compound Identification

In this chapter, we present pilot studies on the [identification](#) of [compounds](#).

The goal of these pilot studies is two-fold. Firstly, using a corpus study, we want to find a **practical definition** of [compounds](#). We inspect various commonly established [linguistic criteria](#) which are described as more or less reliable in linguistics literature. The outcome of this inspection provides new insights into the notion of [compoundhood](#). Secondly, we aim to develop a [compound identification](#) method, for which we need easy-to-implement and manual-resource-lean [criteria](#) that are most reliable from a practical point of view.

## 10.1. Linguistic Criterion Inspection

The first pilot study is the [Linguistic Criterion Inspection \(LCI\)](#). We have a look at the [linguistic criteria \(LCs\)](#) for [compoundhood](#) presented in previous linguistics literature, some of which are discussed in Chapter 4. Our immediate aim is to get an idea of the importance and relevance of some [LCs](#) for the determination of the [compoundhood status](#) of a [target](#) expression. Moreover, we try to get an impression of the agreement of the [LC](#) rating, i.e., how much (divergent) subjectivity influences the judgement of [LCs](#)?

### 10.1.1. Europarl Nominal Compoundhood Rating Gold Standard

As concluded in Chapter 6, we adopt the opinion about [compoundhood](#) saying that there is no concrete class ‘[compound](#)’ but there are constructions which can be considered more or less compoundlike.

For the [LCI](#), given an English [word](#) sequence, we want to know in how far the [compoundhood status](#) correlates with a positive indication of the discussed [LCs](#). For this purpose, we need ratings for [compoundhood](#) and for each [LC](#). As grounding source, we

take the [parallel](#) EUROPARL, described in Chapter 9. Subsequently, the resulting gold standard will be referred to as [Europarl Nominal Compoundhood Ratings \(ENCR\)](#).

### Linguistic Criterion Selection

For an objective selection of [linguistic criteria](#), we decided to restrict to the closed set of [LCs](#) discussed by Lieber and Štekauer (2009, chap. 1). These [criteria](#) are:

- 1. Spelling:** The orthographical [criterion](#) presented in Section 4.3. An English [compound](#) can be realized as [closed](#), [hyphenated](#) or [open compound](#), whereas phrases are usually written in several [words](#).
- 2. Inseparability:** The first syntactic [criterion](#) described in Section 4.6. No elements can be inserted between [modifier](#) and [head](#) of a [compound](#), while this is often possible for phrases.
- 3. Inability to modify the [modifier](#):** The second syntactic [criterion](#) described in Section 4.6. For [compounds](#), the [modifier](#) is not able to be modified, whereas this is possible for syntactic phrases (e.g., [NPs](#)).
- 4. Inability to replace the head by the pronoun *one*:** The third syntactic [criterion](#) described in Section 4.6. The [head](#) of a [compound](#) cannot be replaced by the pronoun *one*, whereas this is often possible for phrases.
- 5. Inflection of the [modifier](#):** The morphological [criterion](#) discussed in Section 4.4. In a [compound](#), the [modifier](#) does not undergo any [word inflection](#) operation (e.g., pluralization) but only the [head](#).
- 6. Prosody:** The prosodic [criterion](#) (Section 4.5) states that in phrases, the [head](#) usually gets the primary stress or all elements have equal stress. In contrast, in a compound the primary stress is commonly on the [modifier](#).

### Guidelines for Compoundhood

In the spirit of more or less compoundlike expressions, the better the [LCs](#) are met for an expression  $\Psi$ , the more compoundlike  $\Psi$  is.

Therefore, the annotators are introduced into all relevant [LCs](#) for [compoundhood](#). In addition, to get an impression of the controversy in literature, they are provided with the first chapter of the **Oxford Handbook of Compounding** (Lieber and Štekauer, 2009, chap. 1).



## 10. Pilot Studies on Compound Identification

Given the guideline that the **compoundhood** of an expression is based on the number of **LCs** being met, the motivation of the **LCI** is that not all **LCs** might be considered equally important for the classification decision.

All guidelines for the annotation are given in Appendix E.

### Annotators

For the decision whether an expression is a **compound** and which of the discussed **linguistic criteria** are the critical factors for the classification, we need experts both with respect to the **target language** (i.e., *English*) and with respect to the phenomenon of **compounding**.

Therefore, we employed two native English-speaking experts in linguistics, in particular in the area of **compoundhood**.

### Annotation Process

As environment of the annotation process, we selected APACHE OPENOFFICE<sup>1</sup> Calc, a free and open-source spreadsheet program.



ID	SENTENCE	COMPOUND	COMPOUND RATING	1. SPELLING	2. INSEPARABILITY	3. MOD-MOD	4. ONE-REPLACEMENT	5. INFLECTION	6. PROSODY	COMMENTS
ep-98-03-10.xml.gz:3221:0:46	We believe that this programme , as changed by our amendments , will play a fundamental role in protecting the health of the citizens of Europe , by significantly reducing the number of deaths and									
ep-08-02-18-019.xml.gz:195:0:38	I consider it to be the European Union 's duty , as a democratic entity , to promote respect for the rights of all the Union 's citizens by initiating European programmes of education and information									
ep-11-05-12-013.xml.gz:1783:0:2	At the same time , we should consider introducing cultural visas for artists and for all those working in the cultural sector .									
ep-06-10-23-021.xml.gz:126:0:24	However , the Commission will bear the issues in question in mind within the framework of the development of our policy in this sector .									
ep-01-09-05.xml.gz:1763:0:83	The plain fact is that the majority of this Parliament - the same Parliament that adopted preventive 'apartheid ' measures against the new , democratically elected Austrian government , that seriou									
ep-08-04-22-016.xml.gz:438:0:5	It is really quite amazing .									
ep-05-09-07.xml.gz:1996:0:17	Indeed , instead of giving it due incentives sometimes we are almost encouraging it to leave Europe .									
ep-00-11-30.xml.gz:78:0:32	Before the Erika sank , many other ships sank as well , with the loss of oil , other cargoes , or tragically , the loss of lives of crew or passengers .									
ep-01-12-11.xml.gz:2135:0:21	Which of the existing mechanisms does the Commission intend to reinforce in order to improve the effectiveness of the Euro-Mediterranean partnership ?									
ep-02-02-27.xml.gz:2096:0:15	At present , promoting social and economic development in these states is undoubtedly the imperative .									
ep-11-04-07-003.xml.gz:439:0:7	Some very valid questions have been asked .									
ep-08-07-10-011-02.xml.gz:55:0:	As a member of the South Asian Association for Regional Cooperation ( SAARC ) delegation of the European Parliament , I have visited Bangladesh several times .									
ep-07-07-11-018.xml.gz:41:0:29	Finally , clear political support must be given to every country that uses a flexibility instrument , whatever it may be , which is not the case in practice .									

Figure 10.1.: OpenOffice spreadsheet for **nominal compound** annotation

The annotation file is a spreadsheet consisting of 11 columns, designed to include the following information:

All columns are color-highlighted as shown in Figure 10.1.

<sup>1</sup>openoffice.org

## 10. Pilot Studies on Compound Identification

- Column A:** a unique system-internal ID, referring to a sentence in a EUROPARL document
- Column B:** the English sentence in which **nominal compounds** are to be searched
- Column C:** the observed **nominal compound**
- Column D:** rating for the observed **nominal compound**
- Column E:** rating for the **linguistic criterion: 1. Spelling**
- Column F:** rating for the **linguistic criterion: 2. Inseparability**
- Column G:** rating for the **linguistic criterion: 3. Inability to modify the modifier**
- Column H:** rating for the **linguistic criterion: 4. Inability to replace the head by *one***
- Column I:** rating for the **linguistic criterion: 5. Inflection of the modifier**
- Column J:** rating for the **linguistic criterion: 6. Prosody**
- Column K:** optional comments for the annotation

The annotators have to extract the **word** sequences (including one-**word** constructions) which show a compoundlike character. In the case of more complex expressions (that have three or more **constituents**), all nested **nominal compounds** have to be listed. For each extraction, the annotator has to provide a rating for the **compoundhood** and for each of the six **LCs**, i.e., in how far they are met. The ratings range between 1 and 3. The extracted **compounds** and their ratings are listed below the EUROPARL sentence, as shown in Figure 10.2.

1	ID	SENTENCE	COMPOUND	COMPOUND RATING	1. SPELLING	2. INSEPARABILITY	3. MOD-MOD	4. ONE-REPLACEMENT	5. INFLECTION	6. PROSODY	COMMENT
293	ep-06-12-14-003.xml.gz:219:0:21	welcome the Commission's proposal to encourage public procurement of clean and efficient vehicles , including using high biofuel blends .	public procurement	1	2	2	2	1	3	1	
294			biofuel blends	2	2	3	3	3	3	2	
295			biofuel	3	3	3	3	3	3	3	
297	ep-07-07-10-021.xml.gz:165:0:34	So our offer could be that we will support a CO2-free coal-fired power plant with our technologies , as a gesture , but in return we can demand that our patent rights be respected .	power plant	3	2	3	3	3	3	3	
298			patent rights	3	2	3	3	3	3	3	

Figure 10.2.: Examples of spreadsheet-based **nominal compound** annotation

The annotation is based on introspection of the annotators instead of naturally occurring data, which would be the ideal setting. Using naturally occurring data for all inspected **LCs** would require a vast amount of **compound**-annotated text and speech data. The creation of such data would exceed the scope of this thesis.

### Experimental Workflow

1. As first step, the annotators **read** the guidelines and the first chapter of the *Oxford Handbook of Compounding*.
2. Next, the annotators passed a **training stage** on a common set of 20 EUROPARL sentences. They
  - a) annotated the training set individually (first iteration)

## 10. Pilot Studies on Compound Identification

- b) got feedback about some formal mistakes they made during the annotation (e.g., not focusing on nominal expressions or ignoring embedded [nominal compounds](#))
  - c) revised their annotations (second iteration)
  - d) checked the annotations of their colleague and discussed disagreements in the extraction
  - e) revised their annotators for the last time (third iteration)
3. In the next step, the annotators labelled several hundreds of EUROPARL sentences **individually**.
  4. Finally, the annotators get a common *agreement set* of 51 EUROPARL sentences. In this final annotation step, we refrained from discussing disagreements between the annotators but aimed to illustrate which [Inter-Annotator Agreement \(IAA\)](#) trained and experienced annotators achieve for the [compound identification](#).

### Final Datasets

Since one of the annotators finished the annotation job earlier and provided various interesting comments on the rating of the [LCs](#), we consider the first annotator to be more dedicated to the job and more sensitive in terms of [compound identification](#) and [LC](#) rating.

Therefore, we decided to compile three dataset versions which are based on the annotator.

Dataset	Sentences	Compound tokens	Avg. tokens / sentence
Annotation1	270	624	2.31
Annotation2	195	346	1.77
Combination	394	824	2.09

Table 10.1.: Size of the final ENCR datasets

1. The first dataset contains only [compound tokens](#) annotated by the first annotator (Annotation1), i.e., both the individual sentences and the annotation of the common sentences.

2. The second dataset contains only **compound tokens** annotated by the second annotator (**Annotation2**), i.e., both the individual sentences and the annotation of the common sentences.
3. The third dataset contains a combination of both annotators (**Combination**), i.e., the individual sentences of both the first and the second annotator. In addition, we decided to include the annotations of the first annotator for the common EUROPARL sentences.

The size of the three datasets is given in Table 10.1.

The last column shows the average number of **identified compound tokens** per investigated EUROPARL sentence. It turns out that the first annotator **identified** more **tokens** (2.31) than the second annotator (1.77), i.e., the first annotator has the more tolerant notion of **compoundhood**.

### 10.1.2. Inter-Annotator Agreement on the ENCR

This section addresses the **Inter-Annotator Agreement (IAA)** of the **ENCR**. The goal of this section is threefold.

1. Firstly, the **IAA** allows us to estimate the agreement in **compound identification**, providing an **UPPER** bound for the **cross-lingual compound identifier**, which will be presented in Chapter 11. Thus, we decided to use the same metrics both for the **IAA** and the evaluation of the **identifier** and not Kappa scores or the like. Another indication for **identification** agreement is the **compoundhood** rating.
2. Secondly, using the **IAA** for the ratings of all proposed **linguistic criteria**, we can get an impression, which **LC** ratings are most reliable, because both annotators agree in the other's rating, and which **LCs** are to be treated with caution, because their ratings are controversial.
3. Finally, we want to illustrate the quality of the extractions and ratings in the **ENCR** gold standard. The higher the agreement of the extractions and of the ratings from all common sentences, the higher the reliability of the annotators and thus the higher quality of the **ENCR** including all individual sentences.

As described in the *experimental workflow* in Section 10.1.1, there are 20 EUROPARL sentences annotated in three iterations in a *training stage* and a final *agreement set* of

51 common sentences which have been annotated individually without discussion on disagreements. We show agreement results for both sets. For the results of the *training stage*, we compare all three iterations.

### Agreement for Compound Identification

In order to measure the IAA for **compound identification**, we look at the **overlap of extracted candidates**. As measure for the overlap, we use the Jaccard coefficient, as given in Formula 10.1, where **Anno- $n$**  refers to the set of **compound** extractions of the  $n$ -th annotator.

$$Jaccard = \frac{|\text{Anno1} \cap \text{Anno2}|}{|\text{Anno1} \cup \text{Anno2}|} \quad (10.1)$$

As discussed above, we consider the first annotator as more dedicated to the annotation job. Thus, we decided to use this person’s data as reference annotation layer and measure precision, recall and  $f_1$ -Score for the **compound** extractions of the second annotator.

The agreement on **compound** extraction in the *training stage* is given in Table 10.2.

Iter	# extractions			Jaccard	Precision	Recall	F <sub>1</sub> -Score
	Anno1	∩	Anno2				
1	38	21	37	0.389	0.568	0.553	0.560
2	39	28	42	0.528	0.667	0.718	0.691
3	45	38	46	0.717	0.826	0.844	0.835

Table 10.2.: Agreement on compound extraction in the *training stage*

The best extraction IAA is achieved after the third training iteration with a Jaccard coefficient of **0.717** and an  $f_1$ -Score of **0.835**.

The agreement on **compound** extraction in the *agreement set* is given in Table 10.3.

# extractions			Jaccard	Precision	Recall	F <sub>1</sub> -Score
Anno1	∩	Anno2				
119	66	100	0.431	0.660	0.555	0.603

Table 10.3.: Agreement on compound extraction in the *agreement set*

Surprisingly, the extraction agreement in the final annotation stage is worse than for the third training iteration. The Jaccard coefficient is **0.431** and the  $f_1$ -Score reaches **0.603**, which is slightly better than for the first training iteration.

## 10. Pilot Studies on Compound Identification

Table 10.4 shows some examples of **compound** candidates that have been **identified** exclusively by one of the annotators. It turned out that the first annotator tends to interpret many **adjective-noun** sequences as **nominal compounds**, which are considered as phrasal by the second annotator. Moreover, the second annotator does not consider the **PoS** of the **head** carefully enough (but extracted pronouns and adjectives) and considered acronyms as **nominal compounds**.

<b>Anno1</b>	<b>Anno2</b>
<i>specific aid</i>	<i>USA</i>
<i>economic policy</i>	<i>international</i>
<i>political forces</i>	<i>a third</i>
<i>statistical data</i>	<i>ourselves</i>
<i>daily crime</i>	<i>PSE</i>

Table 10.4.: Exclusive extractions for both annotators

Therefore, we can conclude that **compound identification** is a non-trivial task and even trained and experienced annotators show strong disagreements in their notion of **compoundhood**. The values for precision, recall and  $f_1$ -Score in the *agreement set* serve as UPPER bound for the subsequent **compound identification** task.

As discussed in Section 8.2.5, Vincze et al. (2011), who manually compiled the Wiki50 corpus, observed an **IAA** with a Jaccard coefficient of 0.552 (0.121 better than our **IAA**) and with precision/recall/ $f_1$ -Score of 0.71 (10.7% better than our **IAA**) for annotating **compounds** in context. Vincze et al. (2011) figured out the same reason for the disagreements, viz., conceptual differences (cf. Table 10.4), “e.g. **hyphenated noun compounds** were not to be marked, however, one annotator occasionally marked phrases like *brother-in-law* as **noun compounds**”.

As additional measure for the **IAA** on **compound identification**, we use the average difference of the **compoundhood rating** on two common sets of EUROPARL sentences, as described in the experimental workflow in Section 10.1.1.

<b>Iter</b>	<b>Average rating of Compoundhood</b>		
	<b>Anno1</b>	<b>Anno2</b>	<b>Difference</b>
1	2.571	1.714	0.857
2	2.571	1.750	0.821
3	2.316	1.763	0.553

Table 10.5.: Average difference in compoundhood rating in the *training stage*

## 10. Pilot Studies on Compound Identification

For the *training stage*, the **compoundhood** rating agreement is given in Table 10.5, and for the *agreement set* in Table 10.6.

Average rating of Compoundhood		
Anno1	Anno2	Difference
2.303	2.076	0.227

Table 10.6.: Average difference in compoundhood rating in the *agreement set*

In contrast to the agreement of **identified compounds**, the rating of **compoundhood** shows a positive effect over time, after the stages of training and individual annotation. Starting with **0.857**, the average difference in the **compoundhood** rating decreases to **0.553** in the third training iteration and further down to **0.227** in the final *agreement set*.

Our final conclusion for the IAA on **compound identification** is that while the **identification** of **compounds** is a non-trivial controversial task even for trained and experienced annotators, rating **compoundhood** for common **compound** candidates seems to be trainable.

### Linguistic Criterion Rating

Iter	Average rating of LC		
	Anno1	Anno2	Difference
Spelling			
1	1.667	2.048	0.381
2	1.107	1.464	0.357
3	1.132	1.526	0.395
Inseparability			
1	2.429	2.857	0.429
2	2.393	2.643	0.250
3	2.289	2.632	0.342
Inability to modify the <b>modifier</b>			
1	2.095	2.905	0.810
2	1.964	2.929	0.964
3	1.868	2.921	1.053

Table 10.7.: Average difference in LC rating in the *training stage* (1)

For the six **linguistic criteria (LCs)** used in the **ENCR**, we check the agreement in terms of average rating difference. For the *training stage*, the average ratings are given

in Tables 10.7 and 10.8, and for the *agreement set*, we show average ratings in Table 10.9.

Iter	Average rating of LC		
	Anno1	Anno2	Difference
Inability to replace the <b>head</b> by <i>one</i>			
1	2.286	2.048	0.238
2	2.357	2.214	0.143
3	2.526	2.368	0.158
Inflection of the <b>modifier</b>			
1	2.286	2.048	0.238
2	2.357	2.214	0.143
3	2.526	2.368	0.158
Prosody			
1	2.190	2.857	0.667
2	2.357	2.786	0.429
3	2.184	2.763	0.579

Table 10.8.: Agreement on LC rating in the *training stage* (2)

As first result, we see that the average LC ratings and differences do not change strongly across the three training iterations. This is to be expected, since with our feedback and the discussion on disagreements (between the iterations), we focused on the candidate **compound identification** rather than the LC ratings. For the final annotation stage, we observed that the average LC rating difference ( $\sim 0.351$ ) is smaller than the average rating difference for the *training stage* ( $\sim 0.447$ ). One reason for this could be that the annotators got self-trained during the individual annotation stage.

When comparing the LC ratings in the final *agreement set* (Table 10.9), it turned out that the spelling **criterion** has the strongest agreement with an average rating difference of **0.030**, followed by the inability to modify the **modifier** (**0.212**) and the inseparability (**0.288**). The **criterion** for which there is least agreement is prosody, with an average rating difference of **0.667**. This is to be expected, since linguistics literature argues that prosody of **compounds** can alter across speakers and dialects (Nakov, 2013).

### 10.1.3. Experiments on the Linguistic Criterion Inspection

After knowing which LC ratings are most reliable with respect to IAA, in the following experiments, we aim to determine which LCs correlate most and least with a positive **compoundhood** judgement.



10. Pilot Studies on Compound Identification

Average rating of LC		
Anno1	Anno2	Difference
Spelling		
1.288	1.318	0.030
Inseparability		
2.303	2.591	0.288
Inability to modify the modifier		
2.470	2.682	0.212
Inability to replace the head by <i>one</i>		
1.909	2.364	0.455
Inflection of the modifier		
1.909	2.364	0.455
Prosody		
1.909	2.576	0.667

Table 10.9.: Agreement on LC rating in the *agreement set*

For the experiments, we use the third final dataset (Combination), comprising **824 nominal compound tokens**, as described in Section 10.1.1.

## Distribution of Ratings

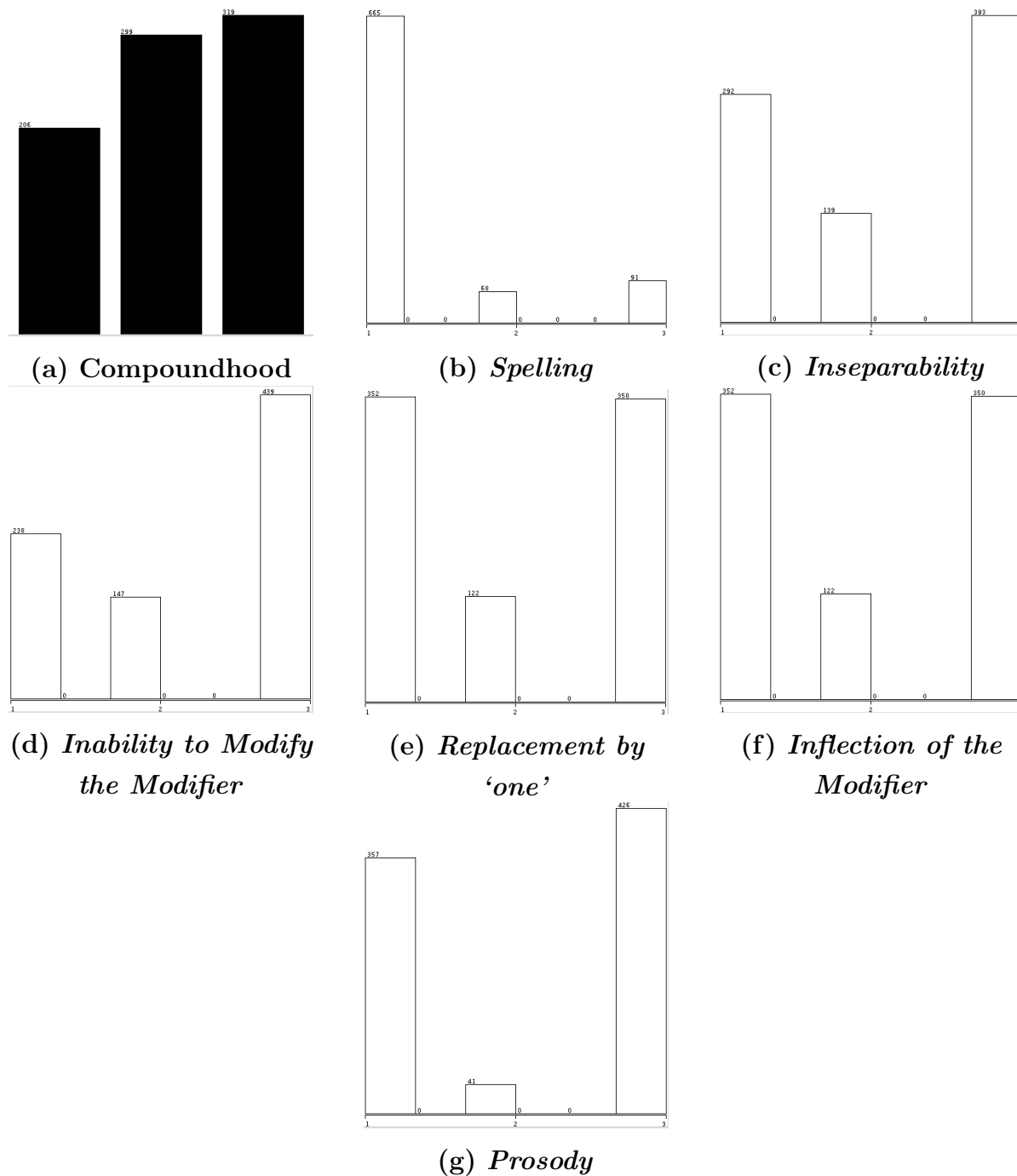


Figure 10.3.: Distribution of Compoundhood and LC Ratings

As discussed in Section 10.1.1, the annotators have to decide between the ratings 1, 2 and 3. Figure 10.3 shows the distribution of these three rating classes for all dataset

samples, as provided by the WEKA toolkit (Hall et al., 2009).

As a first observation, we can conclude that the **compoundhood** rating (Figure 10.3(a)) can be considered as graded. If a **word** sequence has a compoundlike character, it is mostly rated as 3, then 2 and least frequently as 1.

Another result is that most **linguistic criteria** (Figure 10.3(c)-(g)) can be considered as a binary class, i.e., 1 and 3 have been selected most time and 2 is a minor class for controversial cases.

As a final observation, we see that the *spelling criterion* (Figure 10.3(b)) has the very dominant class 1, whereas both 2 and 3 are very infrequent. This is to be expected, since English is considered as an **open compounding language**, i.e., most **compounds** consist of several **words**.

## Classification Experiments

In order to determine which **linguistic criteria** are most and least relevant for the **compoundhood** decision, we perform some classification experiments. All subsequent classification methods are provided by the WEKA toolkit.

**Decision Tree Classification** As a popular way for determining crucial features in supervised machine learning, we employ a J48 decision tree classifier (Quinlan, 1993) with a 10-fold cross-validation.

As baseline, we select the majority class (rating 3) baseline, having a prediction accuracy of **38.7%**.

No.	LC Selection	Accuracy
1	<i>Spelling</i>	45.1%
2	<i>Inseparability</i>	50.7%
3	<i>Inability to Modify the Modifier</i>	50.1%
4	<i>Replacement by ‘one’</i>	48.2%
5	<i>Inflection of the Modifier</i>	48.2%
6	<i>Prosody</i>	48.9%
	all LCs	<b>63.7%</b>

Table 10.10.: Compoundhood prediction accuracy using a J48 decision tree

While the best **LC** selection includes all **LCs**, achieving an accuracy of **63.7%**, the best single **LC** for the J48 decision tree is the **criterion** for the *Inseparability* (50.7%), followed by the *Inability to Modify the Modifier* (50.1%) and *Prosody* (48.9%).

## 10. Pilot Studies on Compound Identification

For measuring **statistical significance**, Table 10.11 presents feature groups (referring to the **LC** number, given in the first column of Table 10.10).

Group ID	LC Numbers	Accuracy
A	1, 2, 3, 4, 5, 6	63.7%
B	2, 3, 4, 5, 6	63.1%
C	2, 3, 6	62.5%
D	2, 3	56.4%
E	2	50.7%
F	1, 4, 5	47.2%

Table 10.11.: Compoundhood prediction using feature groups in a J48 decision tree

In Table 10.12, we show for each pair of feature group, whether there is a statistically significant difference in compoundhood prediction accuracy (as given in Table 10.11), based on the z-test for proportions, with a significance level of  $p < 0.05$ .

E	YES				
D	YES	YES			
C	YES	YES	YES		
B	YES	YES	YES	NO	
A	YES	YES	YES	NO	NO
	F	E	D	C	B

Table 10.12.: Statistical significance test for a J48 decision tree

Subtracting the least precise **LCs** (A to B, B to C or A to C) does not lead to a significant difference in performance. However, removing any of the most precise **LCs** (C to D, D to E) worsens the performance significantly. Finally, comparing the group of the three most precise **LCs** (C) with the group of the least precise **LCs** (F), we observe a statistically significant difference.

Figure 10.4 shows the decision tree visualization that achieves the highest accuracy. The *inseparability criterion* is positioned on the **root node**. For the value 1, the **LC** for the *inability to modify the modifier* (modmod) is most decisive, whereas for the values 2 and 3, the most decisive **criterion** is *spelling*, the worst performing **criterion** in Table 10.10. A possible reason for this is that it is hard to distinguish between the ratings 2 and 3.

Thus while the unreliable *spelling criterion* is located on the second level of the decision tree, it does not help a lot to distinguish between the ratings 2 and 3. We will inspect the decision tree for a binary classification (ratings 1 vs. (2 or 3)) in future work.

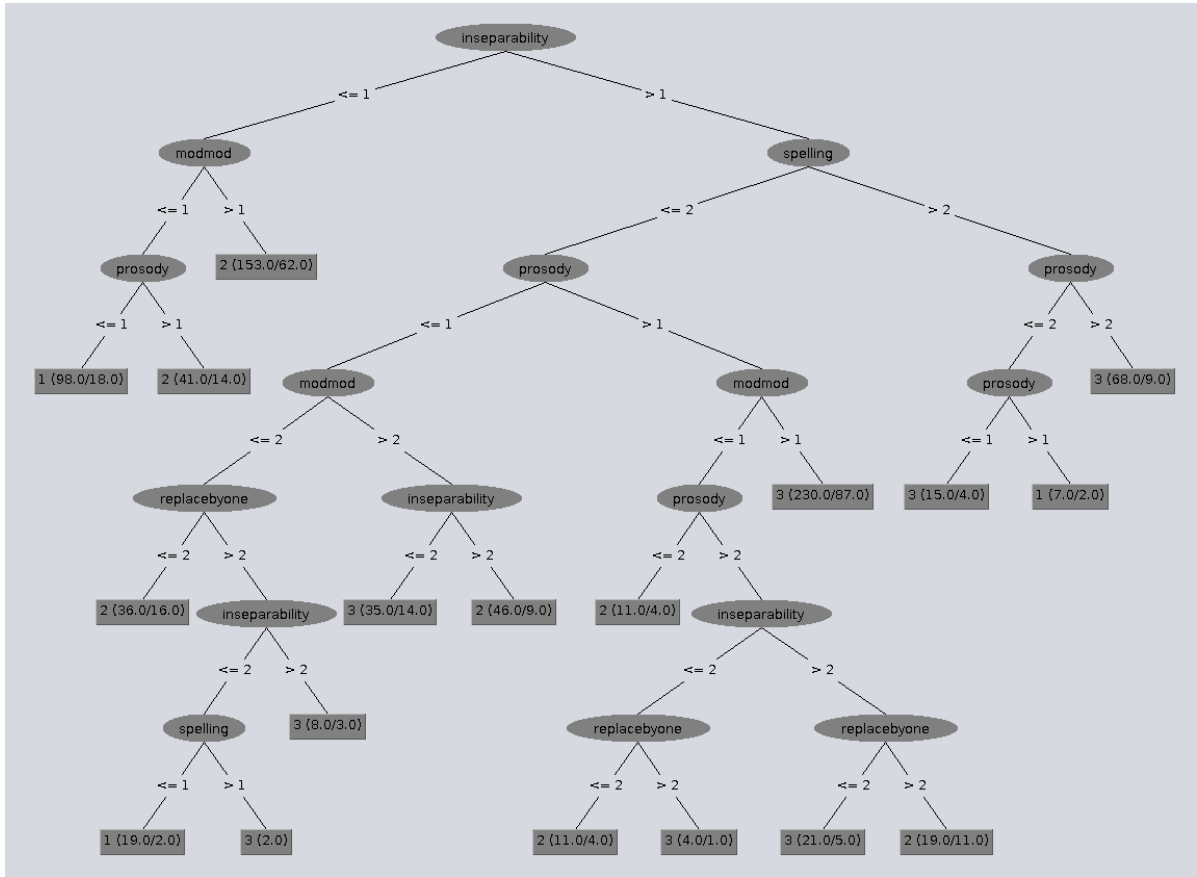


Figure 10.4.: The J48 decision tree for all linguistic criteria

**Naive Bayes Classifier** As alternative classifier, we performed a 10-fold cross-validation on a Naive Bayes classifier. The classification accuracies are given in Table 10.13.

The overall result for using Naive Bayes is the same as for using the J48 decision tree classifier: the best *LC* selection includes all *LCs*, achieving with an accuracy of **55.7%**. The best single *LC* for Naive Bayes is the *criterion* for the *Inability to Modify the Modifier* (50.1%), followed by the *Inseparability* (48.9%) and *Prosody* (48.9%).

In analogy to the J48 Decision Tree classification, for measuring **statistical significance**, Table 10.14 presents feature groups (referring to the *LC* number, given in the first column of Table 10.13).

10. Pilot Studies on Compound Identification

No.	LC Selection	Accuracy
1	<i>Spelling</i>	43.8%
2	<i>Inseparability</i>	48.9%
3	<i>Inability to Modify the Modifier</i>	50.1%
4	<i>Replacement by 'one'</i>	48.2%
5	<i>Inflection of the Modifier</i>	48.2%
6	<i>Prosody</i>	48.9%
	all LCs	<b>55.7%</b>

Table 10.13.: Compoundhood prediction accuracy using a Naive Bayes classifier

Group ID	LC Numbers	Accuracy
A	1, 2, 3, 4, 5, 6	55.7%
B	2, 3, 4, 5, 6	57.4%
C	2, 3, 6	55.6%
D	3	50.1%
E	1, 4, 5	44.1%

Table 10.14.: Compoundhood prediction using feature groups in a Naive Bayes classifier

In Table 10.15, we show for each pair of feature group, whether there is a statistically significant difference in compoundhood prediction accuracy (as given in Table 10.14), based on the z-test for proportions, with a significance level of  $p < 0.05$ .

D	YES			
C	YES	YES		
B	YES	YES	NO	
A	YES	YES	NO	NO
	E	D	C	B

Table 10.15.: Statistical significance test for a Naive Bayes Classification

Subtracting the least precise LCs (A to B, B to C or A to C) does not lead to a significant difference in performance. However, removing two of the most precise LCs (C to D) worsens the performance significantly. Finally, comparing the group of the three most precise LCs (C) with the group of the least precise LCs (E), we observe a statistically

significant difference.

### Average Difference between Compoundhood and Linguistic Criterion

Table 10.16 shows the average difference between the average **compoundhood** ratings and the different average **LC** ratings for the final dataset.

Average rating of <b>Compoundhood</b> / <b>LC</b>		
<b>Compoundhood</b>	<b>LC</b>	<b>Difference</b>
2.137	Spelling	
	1.303	0.834
	Inseparability	
	2.123	0.014
	Inability to modify the <b>modifier</b>	
	2.244	0.107
	Inability to replace the <b>head</b> by <i>one</i>	
	2.000	0.137
	Inflection of the <b>modifier</b>	
	2.000	0.137
Prosody		
2.084	0.053	

Table 10.16.: Average difference between compoundhood and linguistic criteria

The results are in line with the observations we made for the classification experiments. The agreement between the **compoundhood** and the **inseparability criterion** is strongest, with the smallest average rating difference of **0.014**, followed by the **prosody criterion** (0.053) and the **criterion** for the Inability to modify the **modifier** (0.107).

#### 10.1.4. Conclusion of the Linguistic Criterion Inspection

##### Summary of the Experimental Results

In Section 10.1.2, we measured the **IAA** between two experienced annotators for the **compound** extraction and for the ratings for **compoundhood** and six **linguistic criteria**. Although informed by linguistics literature and by a set of **linguistic criteria**, and although disagreements in the annotations were discussed in a training stage, the annotators decided for diverse individual annotation policies. This discrepancy leads to a small overlap of **compound** extractions. The strongest rating agreement is achieved for

the *spelling criterion*, followed by the inability to modify the *modifier* and inseparability (Table 10.9).

For the LCI experiments (10.1.3), we applied a decision tree and a Naive Bayes classifier on the task of predicting the *compoundhood* rating given the LC ratings. We observed that the most decisive LCs are the *inability to modify the modifier*, the *inseparability* and the *prosody* (Table 10.10). This result is in line with an experiment on the average rating agreement between *compoundhood* and the six LCs (Table 10.16).

### Conclusion for the Notion of Compoundhood

The three *linguistic criteria* that correlate best with *compoundhood* are (1) *inseparability*, (2) *inability to modify the modifier* and (3) *prosody*.

In conclusion, we propose the following indicators for characterizing English *compounds*. Since these indicators are neither necessary nor sufficient, they should be considered as in a graded scale, i.e., the *compoundhood* level tends to rise as more indicators are satisfied.

An English word sequence  $\Psi$  tends to be a compound if

- (1) no element (e.g., an adjective) can be inserted in  $\Psi$  with preserving the meaning
- (2) the non-final *constituents* of  $\Psi$  cannot be modified by external *words*
- (3) one of the non-final *constituents* of  $\Psi$  gets a prosodic stress

### Conclusion for the Compound Identification

The results in the IAA experiments revealed that the *identification* of *compounds* is a non-trivial task, and even trained and experienced annotators show significant disagreements. For evaluating our proposed *compound identification* method, we use as UPPER bound a Jaccard coefficient of **0.431** and an  $f_1$ -Score of **0.603**.

An ideal way to exploit the results of the LCI experiments for the *compound identification* would be the implementation of the *linguistic criteria* that correlate most with *compoundhood*. However, in the spirit of avoiding manual resources and relying on *indirect supervision*, we do not find an approach to implementing these *criteria* which is in line with our methodological key concepts:

- The *inseparability criterion* cannot be implemented in a *type*-based fashion (e.g., monolingual), since even if we see many more consecutive than discontinuous *types*, it could still be the case that there is a phrasal and a *compound* version of the



**target** (e.g., *French teacher*). The **LC** cannot be implemented using **cross-lingual supervision**, because there might be phrasal translations for which intervening elements do not indicate that the **target** is separable. For example, the English **2NC** *labour market* modified by the prenominal adjective *European* could be realized in Italian as *mercato europeo del lavoro* (lit: ‘market European {of the} labour’).

- For the *inability to modify the modifier criterion*, a possible **cross-lingually supervised** implementation could be the morphological agreement in Romance languages. For determining the **compoundhood status** for  $N_1N_2$  in the English **word** sequence  $ADJ N_1 N_2$ , we could use the French phrasal equivalent  $\tau(N_2)$  de  $\tau(N_1)$   $\tau(ADJ)$ , where  $\tau(\psi)$  refers to the French **equivalent** of  $\psi$ . If there is a morphological disagreement between  $\tau(adj)$  and  $\tau(N_2)$  (e.g., in gender) but an agreement between  $\tau(adj)$  and  $\tau(N_1)$ , we would have an indication for an English phrase. However, there are many cases in which we cannot find such combinations of morphological agreement and disagreement due to the sparseness issue in **parallel corpora**. Finally, there is need for morphological tags, which requires a profound and language-specific morpho-syntactic analysis, often based on manual resources.
- For the *prosody criterion*, to the best of our knowledge, there is no large-scale spoken corpus annotated with prosody.

The **linguistic criterion** that correlates least with **compoundhood**, i.e., the *spelling criterion*, has the strongest **IAA**. The reason for the poor correlation between *spelling* and **compoundhood** is obvious: English is an **open compounding language** and realizes **compounds** as multi-word constructions resembling phrases. However, knowing from linguistics literature that the *spelling* is a much more decisive **criterion** for other languages (e.g., **closed compounding languages**), in the next section, we will inspect the behavior of English **compounds** across languages with respect to *spelling*.

### Limitations and Future Work

We are aware of the fact that the experiments in the **LCI** are not complete. We only focused on candidate **compounds** with a minimum **compoundhood** rating of 1. For a more representative correlation between **compoundhood** and the **linguistic criteria**, we also need **LC** ratings for **word** sequences having a **compoundhood** rating of 0. In future work, we will add samples of not extracted **word** sequences (stipulating a **compoundhood** rating of 0) and let annotators rate for the six **linguistic criteria**.

As discussed in Section 10.1.1, the annotation is based on introspection of the annotators instead of naturally occurring data. Using naturally occurring **compound**-annotated data would provide context-dependent **LC** ratings that are independent of the annotation task.

## 10.2. Cross-lingual Compound Inspection

The **linguistic criteria** that correlate most with **compoundhood** cannot be implemented in a manual-resource-lean manner and based on **indirect supervision**, as concluded in Section 10.1.4. While the *spelling criterion* correlates least with **compoundhood** in English, this **LC** is much more decisive for other languages, as discussed in linguistics literature.

Therefore, we decided to have a look at **compounding** from a **cross-lingual** perspective, as given in our **parallel corpus** (Chapter 9). In addition to the *qualitative* discussion in Chapter 5, this experiment addresses a *quantitative* study on the **cross-lingual compounding**. The goal of the **Cross-lingual Compound Inspection (XCI)** is to get an impression of how English **compounds** are realized in other languages, where this realization of **cross-lingual** equivalents is restricted to the surface form, which is represented as **universal surface patterns (USPs)**, i.e., a generalized variant of **PoS Patterns**, described in Appendix A.

### 10.2.1. Compound Resource

Since the definition of **compounds** is controversial (as discussed in Chapter 4) and we do not want to stipulate a grounding definition for the underlying experiments, we decided to exploit existing **compound** resources. There are various types of such resources, as has been discussed in Section 8.2. For the **XCI**, we selected the English **2NC** resource developed by Ó Séaghdha (2007), which will be subsequently called **OS2007GS**. The **OS2007GS** resource contains 1443 English **2NCs**, from which 468 **compound** types (and 11,793 **compound** tokens) can be found in **EUROPARL**.

### 10.2.2. Frequency Distributions for Aligned USPs

We consider the **USPs** in all aligned languages and various language families, resulting from automatic **word alignment** to the English **compound** samples in **OS2007GS**. Therefore, we accumulate all **USPs** into frequency distributions.

## 10. Pilot Studies on Compound Identification

Firstly, all aligned languages are considered. The most frequent **USPs** (with a percentage above 1%) are given in Table 10.17.

USP	Frequency	Percentage
CN	24,487	29.7%
SN FC SN	22,173	26.9%
SN	9357	11.4%
SN SN	9347	11.3%
SN ADJ	8491	10.3%
ADJ SN	3765	4.6%

Table 10.17.: **XCI - USP** frequency distribution for all aligned languages

The most frequent **USP** among all aligned languages is the **closed nominal compound** (CN) with 29.7%. The second most frequent **USP** is the **complex nominal**, i.e., a simplex noun followed by a sequence of **function words** (e.g., prepositions) and finally another simplex noun (SN FC SN), with 26.9%. The third most frequent **USP** (11.4%) is a simplex noun (SN). Besides **undersplitting** of **closed compounds** and **word alignment errors**, another possible reason for **atomic** equivalents of English **2NCs** is the phenomenon of asymmetric translations, as discussed in Section 5.3. Altogether (CN + SN), 41.1% of the **USPs** aligned to an English **2NC** are single **words**.

Secondly, we have a look at the aligned **USPs** in all Germanic **closed compounding languages**, i.e., *Danish*, *Dutch*, *German* and *Swedish*. Table 10.18 shows the most frequent Germanic **USPs** (with a percentage above 1%).

USP	Frequency	Percentage
CN	24,487	70.6%
SN	5735	16.5%
SN SN	1525	4.4%
ADJ SN	1129	3.3%
ADJ CN	559	1.6%

Table 10.18.: **XCI - USP** frequency distribution for Germanic languages

By a wide margin, the most frequent aligned **USP** among the Germanic **closed compounding languages** is the **closed nominal compound** (CN, with 70.6%), followed by a simplex noun (SN) with 16.5%. Altogether (CN + SN), 87.1% of the Germanic **USPs** aligned to an English **2NC** are single **words**.

## 10. Pilot Studies on Compound Identification

Thirdly, we have a look at the Hellenic language familie, i.e., on *Greek*. Table 10.19 shows the most frequent Greek **USPs** (with a percentage above 1%).

USP	Frequency	Percentage
SN SN	5533	53.7%
ADJ SN	2418	23.5%
SN	887	8.6%
SN FC SN	599	5.8%
FC SN SN	136	1.3%

Table 10.19.: **XCI - USP** frequency distribution for Greek

Similar to *English*, *Greek* also produces many **open compounds**: 53.7% of the English **2NCs** are realized as **2NC** (SN SN) in *Greek*. In a closer inspection, we observed that the majority of the Greek cases with SN SN are true **2NCs**, followed by **word alignment errors** (e.g., a missing **function word** between the simplex nouns) and **PoS tagging errors**, i.e., the first simplex noun should have been tagged as adjective, resulting in the second most frequent Greek **USP** ADJ SN, with 23.5%.

Finally, we look at the Romance language familie, i.e., on the four languages *Spanish*, *French*, *Italian* and *Portuguese*. Table 10.20 shows the most frequent Romance **USPs** (with a percentage above 1%).

USP	Frequency	Percentage
SN FC SN	21,420	57.3%
SN ADJ	8450	22.6%
SN	2735	7.3%
SN SN	2289	6.1%

Table 10.20.: **XCI - USP** frequency distribution for Romance languages

By a wide distance, the most frequent aligned **USP** among the Romance **open compounding languages** is the **complex nominal** (SN FC SN) with 57.3%. The second most frequent Romance **USP** is the sequence of simplex noun followed by a post-nominal adjective (SN ADJ) with a percentage of 22.6%. Altogether, these two **USPs** represent the majority of 79.9%. Their prominence is also reflected in the fact that there are English **2NCs** that can be realized in both ways in a Romance language, for example, the English *death penalty* can be realized in *French* as *peine de mort* and as *peine capitale*. The third most frequent **USP**, SN, is a simplex noun. In a closer inspection, it

turned out that in most cases, the [word aligner](#) missed the [head](#) or [modifier](#) from a more frequent [USP](#). The main reason for such [word alignment errors](#) are complex sentence constructions, e.g., asymmetric coordinations. For example, in the English coordination *minimum {food hygiene} and safety standards*, the [2NC](#) *food hygiene* is aligned to the French [head](#) *hygiène* (instead of *hygiène alimentaire*) due to the French coordination *normes minimales d' {hygiène} et de sécurité alimentaire* (lit: ‘standards minimum of hygiene and of safety food<sub>ADJ</sub>’).

### 10.2.3. Conclusion of the Cross-lingual Compound Inspection

#### Summary

In the [XCI](#), we observed that [closed nominal compounds](#) (CN) are the most frequent [cross-lingual](#) realization of an English [2NC](#), followed by [complex nominals](#) (SN FC SN). Different language families show different frequency distributions of aligned [USPs](#). While Germanic [closed compounding languages](#) mostly realize English [2NCs](#) as [closed compounds](#) or simplex nouns, *Greek* creates [2NCs](#) and Romance languages create [complex nominals](#) or sequences of simplex nouns and post-nominal adjectives.

#### Conclusion for the Compound Identification

In the following method for [cross-lingual compound identification](#), we exploit the observation that there are frequently [closed cross-lingual equivalents](#) English [compounds](#) in the four Germanic [closed compounding languages](#).

#### Limitations and Future Work

We are aware of the fact that the experiments in the [XCI](#) are not complete. For a more representative result, we would need [cross-lingual](#) frequency distributions of [USPs](#) for true bipartite non-[compounds](#) (e.g., phrasal adjective-noun sequences). We would need resources of true non-[compounds](#) to be matched with EUROPARL. This completion with non-[compounds](#) will be addressed in future work.

Moreover, the current experiment in [XCI](#) only considers the majority class of [nominal compounds](#), [2NCs](#). We expect to see a smaller degree of [cross-lingual closed compounding](#) for English [nominal compounds](#) with three or more [constituents](#). For now, we cannot find [compound](#) resources providing enough [nominal compounds](#) with three or more [constituents](#) for a usable overlap with EUROPARL. The extension to [compounds](#) with more

## 10. *Pilot Studies on Compound Identification*

than two [constituents](#) or other [word](#) category combinations (e.g., adjective-noun [compounds](#)) will be addressed in future work.

# 11. Compound Identification Method

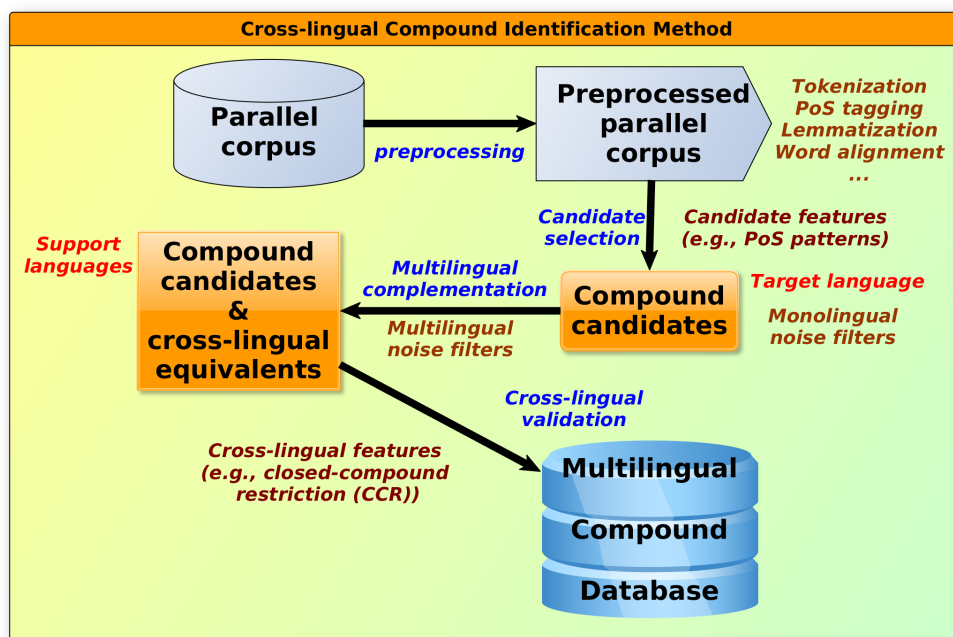


Figure 11.1.: Cross-lingual Compound Identification Method

Figure 11.1 shows the workflow of the [cross-lingual compound identification](#) method. The starting point is a [parallel corpus](#) with a [target language](#) (e.g., English) and some parallel [support languages](#) (e.g., German). For the subsequent experiments, we used a preprocessed part of EUROPARL comprising 10 languages, as described in Chapter 9.

## 11.1. Compound Candidate Selection

From the preprocessed [parallel corpus](#), we extract all plausible [compound](#) candidates of the [target language](#). As observed in the LCI (Section 10.1), the **spelling** criterion in terms of a single [closed compound](#) (4.3) is least reliable for the determination of a

**compoundhood status**, because there is a variety of possible realizations of an English **compound**.

As a common preselection step in **MWE discovery** (Ramisch et al., 2010c), we decided to use the **spelling** criterion for the **compound** candidate selection. For modelling all plausible surface forms of **compounds**, we use a set of predefined **PoS patterns** and complexity information about single nouns, i.e., whether a single noun is a **closed compound**. A **closed** or **hyphenated compound** corresponds to the **PoS pattern** NN. There are various possible combinations of **PoS** for **open compounds**. The set of **PoS patterns** for modelling English **open compounding** is motivated by linguistic discussion about English **compounding** (as in Section 3.9.1) and corpus observations. Table 11.1 lists some motivating examples of **binary** and **ternary nominal compounds**. For all examples, we observed **closed compounds** in German, e.g., *overall recovery rate* is translated to German as *Gesamtrückforderungsquote*. From all plausible **PoS patterns** for **closed** and **open compounds**, we defined two regular expressions generalized for covering all possible combinations up to a **compound size** of 10 **constituents**<sup>1</sup> (last two rows of Table 11.1). As **PoS** tag set, we use the Penn Treebank tag set (Marcus et al., 1993).

<b>PoS pattern</b>	<b>Example</b>
<b>Binary nominal compounds</b>	
NN	<i>marketplace</i>
NN NN	<i>death penalty</i>
JJ NN	<i>structural policy</i>
NN POS NN	<i>children's development</i>
<b>Ternary nominal compounds</b>	
NN NN NN	<i>energy security goal</i>
JJ NN NN	<i>overall recovery rate</i>
Regular expressions for <b><i>k</i>-ary nominal compounds</b> ( $2 \leq k \leq 10$ )	
NN (POS? NN){0,9}	
JJ NN (POS? JJ? NN){0,8}	

Table 11.1.: Predefined **PoS patterns** for the **compound** candidate selection

For single nouns (e.g., *marketplace*), we use the complexity information derived from the rudimentary **compound** splitter, described in Section 9.2. If the noun is split into two **constituents**, it is kept, otherwise discarded.

<sup>1</sup>While the largest plausible **word** sequence observed in Europarl comprised 7 **words**, we defined an upper bound of 10 **words**.



## Monolingual Noise Filters

Afterwards, the **compound** candidates have to be filtered from noise which is due to the automatic preprocessing tools applied to the **parallel corpus** (9.2), e.g., **PoS** tagging errors. With increasing **word** sequence length, the amount of noise increases. We apply several filters to each **word** sequence and keep only those that pass all filters.

**(a) Non-constituent Filter** We disqualify **word** sequences including nouns or adjectives that (1) consist of only one character or (2) are contained in a stop list<sup>2</sup>.

**(b) Constituent Probability** To account for **PoS** tagging errors, we collect all **words** and their **PoS** tags in the **parallel corpus**. For each **word**, we compute the probability of being tagged as a noun or adjective as given in Formula 11.1.

$$P(\textit{noun/adj} \mid \textit{word}) = \frac{f((\textit{noun} \cup \textit{adj}) \cap \textit{word})}{f(\textit{word})} \quad (11.1)$$

We disqualify English **word** sequences, if they contain a noun or adjective  $w$  with  $P(\textit{noun/adj} \mid w) < \theta$ . After testing several values for  $\theta$ , we have decided to choose  $\theta = 0.15$  because it has turned out to be a promising trade-off between coverage and precision (e.g., accepting **words** like *human* but rejecting **words** like *anywhere*).

## 11.2. Multilingual Complementation

In this step, we add to each **compound** candidate in the **target language** all **cross-lingual equivalents** in all aligned **support languages**. This **multilingual complementation** is dependent on the quality of the automatic sentence and **word** alignment tools from the preprocessing step (9.2). In order to remove noise, we apply some **multilingual filters** which are motivated by our **cross-lingual** observations concerning **compound** translation types, outlined in Chapter 5.

For the **Europarl Nominal Compound Database (ENCD)**, we used the following four **cross-lingual** filters.

**(a) Non-constituent Filter** As already done for the English **compound** candidate selection (11.1), we apply the non-**constituent** filter to all aligned languages.

---

<sup>2</sup>[ranks.nl/stopwords](http://ranks.nl/stopwords)

**(b) Truncation of Extraneous Words** We truncate extraneous words (i.e., determiners, prepositions, verbs and adverbs) from the border of the word sequence. Here, we include knowledge about the headedness in the respective languages: adjectives are removed from the right border for Germanic languages and from the left border for Romance languages.

**(c) Phrasal Treatment** In the first step of this filter, we have to determine, whether the aligned word sequence can be considered as a phrase (e.g., an NP) or whether the aligned words are spread across the complete sentence without forming a syntactic unit. In order to avoid the usage of parsing tools, we use a heuristic. For two adjacent English constituents  $w_i$  and  $w_j$ , we check the distance of the constituent equivalents in an aligned support language, i.e., the smallest pairwise distance between all words in the corresponding aligned word sets (AWSs), i.e., between words of  $AWS(w_i)$  and  $AWS(w_j)$ . If this aligned word distance (AWD) is larger than  $\phi$  words<sup>3</sup>, we do not consider the aligned word sequence as a phrase. If the smallest context between the words in  $AWS(w_i)$  and  $AWS(w_j)$  includes content words, the word sequence is also disqualified as a phrase.

For example, for the 3NC *human<sub>1</sub> rights<sub>2</sub> violations<sub>3</sub>*, we observed the following aligned sentence fragment in Italian: *... che le violazioni<sub>3</sub> gravi e sistematiche dei diritti<sub>2</sub> umani<sub>1</sub> ...* ‘...that serious and systematic violations<sub>3</sub> of human<sub>1</sub> rights<sub>2</sub> ...’. In this Italian fragment, the equivalents for *violations* and *rights* are more than  $\phi = 3$  words apart and are thus not a phrase. Moreover, the smallest context between  $AWS(\textit{rights})$  and  $AWS(\textit{violations})$  contains content words.

If the word sequence is qualified as a phrase, we add determiners and prepositions that occur in the context between  $AWS(w_i)$  and  $AWS(w_j)$ . Disqualified word sequences remain unchanged.

**(d) Non-noun Filter** We remove the word sequence if it does not contain at least one noun.

---

<sup>3</sup>When analysing many instances of Romance phrases aligned to an English nominal compounds, we observed that  $\phi = 3$  is the maximum token distance two nominal constituents can be apart (usually separated by preposition or preposition+determiner).

### 11.3. Cross-lingual Validation

In the last step of the **identification** method, we select the final **compounds** to be extracted from the **parallel corpus**. For this purpose, we exploit **cross-lingual** information as a second type of **identification** features. These features are motivated by the outcome of the **XCI**, discussed in Section 10.2. If a **word** sequence in the **target language** has a plausible translation pattern which is a characteristic of a **compound** translation, the **word** sequence can be considered as a **cross-lingually** validated **compound**, otherwise the **word** sequence is not selected for the final set of **compounds**.

For the **ENCD**, we restrict the **cross-lingual** validation to the **Closed Compound Restrictor (CCR)**, defined as follows:

**CCR(*n*)**: *An English compound candidate is considered to be a nominal compound if it is represented as a single noun in at least **n** languages among the closed compounding support languages.*

The **CCR** is defined in a way that it includes asymmetric translations (5.3) in which a **compound** has an **atomic equivalent** in another language (e.g., *blackbird* translated to German as *Amsel*). Given a **parallel corpus** with  $n > 1$  **closed compounding languages**, this definition leaves space for investigating the optimal degree of **cross-lingual closed compounding** ( $\Xi_{closed}$ ) which is necessary for optimizing the **identification** quality. Because the **cross-lingual** filters, described in Section 11.2, still leave some **word** alignment errors (i.e., English **word** sequences that are aligned to only a part of the true translation), a single **closed compounding language** that realizes the English **word** sequence as a single noun (i.e.,  $\Xi_{closed} = 1$ ) might not be restrictive enough. The **CCR** with  $\Xi_{closed} \geq i$  includes only English **compound** candidates that are aligned to at least  $i$  single nouns in the aligned **closed compounding languages** (i.e., **CCR(*i*)**). We expect that  $\Xi_{closed} = j$  with a high value of  $j$  is too restrictive, because there are true English **compounds** that are realized in **closed compounding languages** as phrases.

For the **ENCD**, we used the four **closed compounding languages** Danish, Dutch, German and Swedish. For testing the optimal value of  $\Xi_{closed}$ , we compiled four versions of the database (i.e., **CCR(1)** to **CCR(4)**). As baseline, we used all **compound** candidates (i.e., **CCR(0)**). We expect **CCR(0)** to have the highest recall for **compound identification**, whereas **CCR(4)** will have the highest precision.

## *11. Compound Identification Method*

# 12. Europarl Nominal Compound Database

As discussed in Chapter 11, we applied our [cross-lingual nominal compound identification](#) method to 10 languages of the [parallel Europarl](#) (including the four Germanic [closed compounding languages](#) Danish, Dutch, German and Swedish), which resulted in the [Europarl Nominal Compound Database \(ENCD\)](#). During the [cross-lingual validation \(11.3\)](#), we used the [CCR](#) feature with the parameter  $\Xi_{closed}$ , i.e., the minimum number of alignments to single nouns among the four Germanic [closed compounding languages](#). Therefore, we have compiled five different versions of the [ENCD](#) with varying value of  $\Xi_{closed}$ : [CCR\(0\)](#) to [CCR\(4\)](#).

## 12.1. Statistics and Cross-lingual Observations in the ENCD

For illustrating the content of the [ENCD](#), we present some interesting statistics and additional [cross-lingual](#) observations.

### 12.1.1. PoS pattern Distribution

In this subsection, we present some statistics about [PoS patterns](#) of English entries in the [ENCD](#) for different values of  $\Xi_{closed}$  ([CCR\(0\)](#) to [CCR\(4\)](#)). We will illustrate that [CCR\(4\)](#) contains more compoundlike entries than [CCR\(0\)](#) does.

Table 12.1 shows the distribution of the top 20 English [PoS patterns](#) for [CCR\(0\)](#), listed in the first column. The following columns show the frequency and ratio of these [PoS patterns](#) for the five different [ENCD](#) versions. The [PoS pattern](#) JJ NN, which is ambiguous between [NPs](#) and [nominal compounds](#) (cf. *French teacher*), is the most frequent pattern for [CCR\(0\)](#) to [CCR\(2\)](#). For [CCR\(3\)](#) and [CCR\(4\)](#), the most frequent [PoS pattern](#) is NN NN, whereas JJ NN is the second or third most common pattern.

12. Europarl Nominal Compound Database

Actually, when we have a look at the most frequent English JJ NN sequences in CCR(0) and CCR(4), it turns out that the latter are much more compoundlike than the first, as shown in Table 12.2. While some JJ NN sequences in CCR(4) are incorrectly PoS-tagged NN NN sequences, others have a relational adjective as *modifier* (e.g., *agricultural policy* or *nuclear energy*). In contrast, most of the adjectives in the English JJ NN sequences in CCR(0) have a functional character (e.g., *same time*, *next item* or *other hand*).

PoS pattern	CCR(0)	CCR(1)	CCR(2)	CCR(3)	CCR(4)
JJ NN	<b>867K (44.7%)</b>	<b>219K (31.5%)</b>	<b>115K (25.3%)</b>	67K (22.4%)	28K (20.2%)
JJ NNS	464K (23.9%)	135K (19.5%)	77K (16.9%)	42K (14.0%)	15K (10.7%)
NN NN	211K (10.8%)	134K (19.3%)	107K (23.6%)	<b>80K (26.9%)</b>	<b>43K (31.0%)</b>
NN NNS	146K (7.5%)	100K (14.4%)	80K (17.7%)	58K (19.6%)	28K (20.6%)
JJ NN NN	54K (2.8%)	11K (1.5%)	4608 (1.0%)	2224 (0.7%)	843 (0.6%)
JJ NN NNS	44K (2.3%)	9717 (1.4%)	4238 (0.9%)	1960 (0.7%)	598 (0.4%)
NN	<b>44K (2.2%)</b>	37K (5.3%)	31K (6.9%)	23K (7.8%)	<b>11K (8.3%)</b>
NNS	<b>19K (1.0%)</b>	18K (2.7%)	16K (3.5%)	13K (4.3%)	<b>6987 (5.1%)</b>
NNS NN	13K (0.7%)	5724 (0.8%)	3786 (0.8%)	2561 (0.9%)	1227 (0.9%)
NN NN NN	13K (0.7%)	6017 (0.9%)	3511 (0.8%)	1847 (0.6%)	671 (0.5%)
NN POS NN	<b>8902 (0.5%)</b>	1319 (0.2%)	500 (0.1%)	170 (0.1%)	<b>56 (&lt;0.1%)</b>
NN NN NNS	8710 (0.4%)	4567 (0.7%)	2889 (0.6%)	1516 (0.5%)	511 (0.4%)
NNS NNS	6366 (0.3%)	3629 (0.5%)	2677 (0.6%)	1766 (0.6%)	816 (0.6%)
JJ NNS NN	4739 (0.2%)	1435 (0.2%)	874 (0.2%)	511 (0.2%)	119 (0.1%)
NNS POS NN	4476 (0.2%)	1373 (0.2%)	861 (0.2%)	516 (0.2%)	221 (0.2%)
NNS POS NNS	4401 (0.2%)	1744 (0.3%)	924 (0.2%)	418 (0.1%)	155 (0.1%)
JJ NN NN NN	3235 (0.2%)	401 (0.1%)	115 (<0.1%)	39 (<0.1%)	5 (<0.1%)
JJ NNS NNS	2964 (0.2%)	1358 (0.2%)	863 (0.2%)	448 (0.2%)	126 (0.1%)
NN POS NNS	2953 (0.2%)	731 (0.1%)	294 (0.1%)	82 (<0.1%)	9 (<0.1%)
JJ NN NN NNS	2158 (0.1%)	293 (<0.1%)	80 (<0.1%)	25 (<0.1%)	4 (<0.1%)
<b>TOTAL</b>	2M	695K	453K	297K	137K

Table 12.1.: PoS pattern distribution in CCR(n)

Another trend we observed in Table 12.1 is that with increasing  $\Xi_{closed}$ , the number of English *closed compounds* increases. While there are 3.2% *closed compounds* (NN + NNS) in CCR(0), there are 13.4% *closed compounds* in CCR(4). Therefore, we can conclude that there is a trend that expressions which are realized as *closed compounds* or *atomic* noun in many aligned languages, are also written as one *word* in English, as already discussed in Section 5.1.1. Some examples observed in CCR(4) are *timetable*, *network*, *background*, *deadline*, *workplace*, *workforce*, *taxpayer* or *weekend*.

A final observation in Table 12.1 concerns the PoS pattern NN POS NN, i.e., two nouns

12. Europarl Nominal Compound Database

CCR(0)	CCR(4)
<i>same time</i>	<i>agricultural policy</i>
<i>internal market</i>	<i>cohesion policy</i>
<i>next item</i>	<i>euro area</i>
<i>other hand</i>	<i>fisheries policy</i>
<i>great deal</i>	<i>security policy</i>
<i>European level</i>	<i>decision-making process</i>
<i>agricultural policy</i>	<i>development cooperation</i>
<i>last year</i>	<i>labour market</i>
<i>economic crisis</i>	<i>nuclear energy</i>
<i>single market</i>	<i>own-initiative report</i>

Table 12.2.: The 10 most frequent English JJ NN sequences in CCR(0) and CCR(4)

with a possessive marker as [linking element](#) (e.g., *children’s song*). While 0.5% of all sequences in CCR(0) are instances of this pattern, there are less than 0.1% in CCR(4). The trend for this pattern is similar to that of JJ NN. Most of these patterns are [NPs](#) (as illustrated for the 10 most frequent sequences with this pattern in Table 12.3) and thus not realized as [closed compounds](#) in the aligned languages. Many [modifiers](#) in CCR(0) are deictic expressions (e.g., *today’s debate* or *tomorrow’s vote*), whereas some expressions in CCR(4) are idiomatic (e.g., *lion’s share*) or contain [modifiers](#) denoting a person (e.g., *women’s movement*).

CCR(0)	CCR(4)
<i>today ’s debate</i>	<i>lion ’s share</i>
<i>today ’s vote</i>	<i>snail ’s pace</i>
<i>tomorrow ’s vote</i>	<i>women ’s movement</i>
<i>rapporteur ’s proposal</i>	<i>prosecutor ’s office</i>
<i>world ’s population</i>	<i>women ’s issue</i>
<i>year ’s budget</i>	<i>citizen ’s initiative</i>
<i>minute ’s silence</i>	<i>auditor ’s report</i>
<i>rapporteur ’s view</i>	<i>world ’s history</i>
<i>today ’s world</i>	<i>world ’s economy</i>
<i>Today ’s vote</i>	<i>women ’s quota</i>

Table 12.3.: The 10 most frequent English NN POS NN sequences in CCR(0) and CCR(4)

The ratios of most other [PoS patterns](#) remain roughly stable across all values of  $\Xi_{closed}$ .

### 12.1.2. Degree of Closed Compounding across Languages

In this subsection, we illustrate how the ENCD can be used for researching differences in [closed compounding](#) across various Germanic languages.

Language	CCR(1)
Danish	421K (60.6%)
Dutch	376K (54.1%)
German	<b>451K (64.8%)</b>
Swedish	335K (48.1%)

Table 12.4.: Degree of [closed compounding](#) among the [closed compounding languages](#) in CCR(1)

In Table 12.4, we have a closer look at the degree of [closed compounding](#) of the four Germanic [closed compounding languages](#) for  $\Xi_{closed} = 1$ , i.e., CCR(1). We selected CCR(1), because it allows for numerous translation types among the [closed compounding languages](#), while not including too much noise (as would be the case for the CCR(0) baseline). The ratio of [closed compounding](#) for each language is determined with the frequency divided by the total size of CCR(1), 695K.

A first observation is that *Swedish* shows the lowest level of [closed compounding](#), i.e., many aligned Swedish expressions are multi-word sequences. For example, the [compound](#) *human rights*, aligned to *Danish*, *Dutch* and *German* as [closed compound](#), is realized in *Swedish* as *mänskliga rättigheter*. The languages that are highest [closed compounding](#) are German and Danish. For example, the expression *first reading* is realized as [closed compound](#) in Danish, *førstebehandlingen*, whereas the other [closed compounding languages](#) realize NPs (e.g., in German *erste Lesung*). The expression *internal market* is an example for a candidate that is realized as [closed compound](#) only in German: *Binnenmarkt*, whereas the other [closed compounding languages](#) create NPs (e.g., in Danish *indre marked*).

### 12.1.3. Paraphrasing and Bracketing 3NCs

As discussed in Section 5.4.3, phrasal translations (5.2) can reveal the [internal structure](#) of a complex [compound](#), e.g., whether a 3NC is LEFT- or RIGHT-branched.

This subsection presents the different PoS patterns for [cross-lingual equivalents](#) of 3NCs in the ENCD, encoded as [universal surface patterns](#) (USPs). The observations described below motivate us to use some USPs for [parsing 3NCs](#). Approaches using



these [Aligned Phrase Patterns \(APPs\)](#) are presented in a pilot study on [compound parsing](#) in Chapter 24.

German	Swedish	French	Italian
CN	CN	SN FC SN FC SN	SN FC SN FC SN
ADJ CN	ADJ CN	SN FC SN ADJ	SN FC SN ADJ
SN FC CN	SN	SN FC SN	SN ADJ
SN	ADJ SN	SN ADJ	SN FC SN
ADJ SN	SN SN	SN SN FC SN	SN SN FC SN
SN SN	SN FC ADJ SN	SN SN SN	SN ADJ FC SN
SN FC ADJ SN	PC CN	SN FC SN SN	SN SN SN
SN SN SN	PC SN	SN ADJ FC SN	SN FC SN SN
CN FC SN	SN SN SN	SN	SN
CN FC CN	SN VB SN	SN SN ADJ	SN SN ADJ

Table 12.5.: The 10 most frequent [equivalents](#) of English [3NCs](#) in [USP](#) format

Table 12.5 shows the 10 most frequent [universal surface patterns \(USPs\)](#) of [3NCs](#) in [CCR\(0\)](#) for four languages: *German*, *Swedish*, *French* and *Italian*. English [3NCs](#) are realized in the [closed compounding languages](#) German and Swedish mostly as [closed compound](#) (e.g., *energy transfer losses* is translated to German as *Energieübertragungsverluste* and to Swedish as *energiöverföringsförluster*). The second most common [USP](#) in the [closed compounding languages](#) is the paraphrasing pattern ADJ CN, i.e., an adjective followed by a [nominal compound](#) as in *trade defence instruments* realized in German as *handelspolitischen Schutzmaßnahmen* (lit: ‘commercial protective measures’). The [USP](#) SN FC CN (i.e., a simple noun followed by a functional context and a [nominal compound](#)) points to a LEFT-branched [3NC](#) (e.g., *energy efficiency legislation* aligned to the German *Gesetzgebung zur Energieeffizienz* (lit: ‘legislation for energy efficiency’)). The lowest [closed compounding](#) language, *Swedish* (see Table 12.4 above), often realizes an English [3NC](#) with the [USP](#) SN FC ADJ SN, as for *human rights policies* aligned to *Swedish* as *politiken de mänskliga rättigheterna*, pointing to a LEFT-branched [3NC](#). An equivalent of this [USP](#) for the Romance languages, usually having a postnominal adjective, is SN FC SN ADJ (the second most common [USP](#) for *French* and *Italian*), for example the English [3NC](#) *car distribution market* is aligned to the French *marché de la distribution automobile*. However, this pattern is slightly ambiguous. In a few cases, the final adjective can also refer to the first noun (i.e., pointing to a RIGHT-branched [3NC](#)). These cases can be detected if there is a declension disagreement (e.g., in gender or number) between the adjective and the respective noun. The counterpart for RIGHT-branched [3NCs](#) is SN ADJ

FC SN, as in *world food supplies* aligned to the French *approvisionnement alimentaire de la planète*. The fact that the LEFT-branched version is much more common than the RIGHT-branched version points to a majority class LEFT for 3NC structures, which will be subsequently used as [LEFT class baseline](#) for the [compound parsing](#) in Part E.

## 12.2. Additional Information

Besides the already presented [word](#) sequences and the corresponding [PoS patterns](#), the [ENCD](#) contains additional information, which can be used for downstream [NLP](#) applications or further research on [cross-lingual compounding](#).

**Context:** [ENCD](#) provides the complete (pseudo) sentence for English and each aligned [support language](#). In these sentences, the relevant [word](#) sequence is highlighted, for example the English sentence “*Therefore, today we are once again calling on all the countries of the world in which the {death penalty} is still used to take immediate measures to abolish it.*” and the aligned German sentence “*Deshalb rufen wir erneut alle Länder der Welt auf, die die {Todesstrafe} immer noch verhängen, sofortige Maßnahmen zu ihrer Abschaffung zu ergreifen.*”.

**Morpho-syntax:** [ENCD](#) integrates the morpho-syntactic tags contained in the OPUS version of Europarl. These tags comprise information about case, number and gender. A possible application is [parsing](#) of English 3NCs using the French [APP SN FC SN ADJ](#), where the adjective can refer to both the first and the second SN. A mismatch between the adjective and a noun in any morpho-syntactic feature can support the disambiguation.

**Lemma sequence:** For each extracted and aligned [word](#) sequence, [ENCD](#) provides both the [word](#) forms and the related [lemmas](#). Besides the context-independent usage of [lemmatized compounds](#), information about [lemmas](#) allows for investigating the degree of pluralization of English [compound modifiers](#), which is used as [linking element](#) in English [noun compounds](#) (3.9.1).

In addition, we stored more information which is necessary for some experiments presented in Part E (e.g., the [word](#) alignment information from the English sentence to any aligned language’s sentence or the [USPs](#) for all extractions).

# 13. Experiments on Compound Identification

In this chapter, we evaluate the [compound identification](#) method proposed in Chapter 11.

## 13.1. Setup

### Data

We applied the [identifier](#) on the [parallel corpus](#), described in Chapter 9, which resulted in the [Europarl Nominal Compound Database \(ENCD\)](#), described in Chapter 12. The [ENCD](#) is represented in five versions corresponding to the degree of [closed compounding](#) of the four Germanic [closed compounding languages](#),  $\Xi_{closed}$ , i.e., [CCR\(0\)](#) to [CCR\(4\)](#).

### Gold Standards

As outlined in Section 10.1.1, there are three versions of a final [ENCR](#) dataset, depending on the annotator (Table 10.1). We provide results on [identifying compounds](#) in the EUROPARL sentences of each [ENCR](#) dataset (i.e., for each [ENCD](#) version, we filter entries belonging to EUROPARL sentences that have not been investigated in [ENCR](#)).

While each [identified compound](#) in [ENCR](#) is associated with a [compoundhood](#) rating, we do not consider the [compoundhood](#) rating in these experiments, but use any [compound token](#) with a [compoundhood](#) rating of at least 1.

### Evaluation Measures

For comparing the [identification](#) quality of our proposed method against the [Inter-Annotator Agreement \(IAA\)](#) of [compound identification](#), described in Section 10.1.2, we use the same metrics:

- number of individually and commonly [identified compound tokens](#)

- resulting Jaccard coefficient
- Precision, Recall and F<sub>1</sub>-Score

## 13.2. Results and Discussion

Tables 13.1, 13.2 and 13.3 show the results for the evaluation of our proposed [compound identification](#) method for the three [ENCR](#) datasets. As UPPER bound, we use the Jaccard coefficient of **0.431** and the F<sub>1</sub>-Score of **0.603** observed for the [IAA](#) between the two annotators, described in Table 10.3.

System	# extractions			Jaccard	P	R	F <sub>1</sub>
	GOLD	∩	SYS				
<a href="#">CCR(0)</a>	624	355	1085	0.262	0.327	0.569	0.415
<a href="#">CCR(1)</a>		174	260	0.245	0.669	0.279	0.394
<a href="#">CCR(2)</a>		124	157	0.189	0.790	0.199	0.318
<a href="#">CCR(3)</a>		82	98	0.128	0.837	0.131	0.227
<a href="#">CCR(4)</a>		41	44	0.065	0.932	0.066	0.123

Table 13.1.: Compound Identification Results for `Annotation1` dataset

Table 13.1 shows the results for the `Annotation1` dataset, including annotations of the first annotator only. The best [identification](#) quality in terms of Jaccard coefficient and F<sub>1</sub>-Score is achieved for the [ENCD](#) version with  $\Xi_{closed} \geq 0$ , i.e., [CCR\(0\)](#). The best precision is achieved for [CCR\(4\)](#) (0.932), whereas the best recall with 0.569 is achieved by the [compound](#) candidate baseline [CCR\(0\)](#). In an error analysis, it turned out that we missed a [PoS pattern](#) covering named entity [constituents](#). Thus, the highest possible recall value is not higher than 0.569. We will enrich the [compound identification](#) method with this missing [PoS pattern](#) in future work. As described in Section 10.1.1, the first annotator has a tolerant notion of [compoundhood](#). This leads to a strong recall drop when increasing  $\Xi_{closed}$ , outweighing the precision gain.

Table 13.2 shows the results for the `Annotation2` dataset, including annotations of the second annotator only. In contrast to the results based on the more tolerant first annotator, the stricter perspective on [compounds](#) of the second annotator provides a smaller recall drop when increasing  $\Xi_{closed}$ . As a consequence, the best [identification](#) quality is achieved for [CCR\(1\)](#) with a Jaccard coefficient of 0.196 and an F<sub>1</sub>-Score of

### 13. Experiments on Compound Identification

System	# extractions			Jaccard	P	R	F <sub>1</sub>
	GOLD	∩	SYS				
CCR(0)	346	151	742	0.161	0.204	0.436	0.278
CCR(1)		87	185	0.196	0.470	0.251	0.328
CCR(2)		59	112	0.148	0.527	0.171	0.258
CCR(3)		42	74	0.111	0.568	0.121	0.200
CCR(4)		17	27	0.048	0.630	0.049	0.091

Table 13.2.: Compound Identification Results for Annotation2 dataset

0.328. In general, the [identification](#) quality of our proposed method is better for the Annotation1 dataset. A crucial reason for this is the fact that the second annotator [identified](#) various [compound](#) candidates that do not match our predefined [PoS patterns](#), e.g., acronyms or non-nominal (e.g., adjectival) expressions, exemplified in Table 10.4.

System	# extractions			Jaccard	P	R	F <sub>1</sub>
	GOLD	∩	SYS				
CCR(0)	824	450	1555	0.233	0.289	0.546	0.378
CCR(1)		230	381	0.236	0.604	0.279	0.382
CCR(2)		163	237	0.182	0.688	0.198	0.307
CCR(3)		111	151	0.128	0.735	0.135	0.228
CCR(4)		52	62	0.062	0.839	0.063	0.117

Table 13.3.: Compound Identification Results for Combination dataset

Table 13.3 shows the results for the Combination dataset, including annotations of both annotators. The performance numbers conform with the average of the individual annotation sets. The best [identification](#) quality is achieved for CCR(1) (Jaccard = 0.236, F<sub>1</sub>-Score = 0.382).

In general, all three tables show a strong recall drop and precision gain when increasing  $\Xi_{closed}$ . In Table 13.1, the recall drop outweighs the precision gain that strongly such that the CCR(0) baseline outperforms all other ENCD versions in terms of Jaccard coefficient and F<sub>1</sub>-Score. For the other two ENCR datasets, the best ENCD version is CCR(1). Thus, we can conclude that the CCR is a promising condition for filtering non-compounds. However, this condition is too restrictive, leading to a strong recall drop. In future work, we will try to mitigate this by adding alternative conditions based on other [linguistic criteria](#).

### 13. Experiments on Compound Identification

As outlined in Section 8.1, to the best of our knowledge, there is no suitable previous work on the **identification** of any class of **nominal compounds**. Previous methods are either restricted to subclasses (e.g., **2NCs**, as described in Section 8.1.1) or to superclasses (e.g., **MWEs**, as described in Section 8.1.5). The only previous methods focussing on any class of **nominal compounds** (presented in Section 8.1.2, e.g., the method proposed by Vincze et al. (2011)) are based on knowledge from WIKIPEDIA and cannot be applied to EUROPARL, which our **cross-lingual** method is relying on.

# 14. Bottom Line of Compound Identification

## 14.1. Summary

In Part C, we aimed to find a practical definition for [compounds](#), explored [cross-lingual equivalents](#) and developed a [cross-lingually supervised](#) method for [compound identification](#).

In [Chapter 7](#), we introduced [compound identification](#). We motivated the necessity for [compound identification](#) and [compound](#) resources ([7.1](#)), discussed some contributions provided along with this part and posed some related research questions ([7.2](#)).

In [Chapter 8](#), we presented an outline of previous related work on the [identification](#) and [discovery](#) of [compounds](#) ([8.1](#)) (e.g., the [identification](#) of [2NCs](#) ([8.1.1](#)) or [closed compounds](#) ([8.1.4](#))), and on different [compound](#) resources ([8.2](#)) (e.g., resources for any [compounds](#) ([8.2.1](#)) or for [2NCs](#) ([8.2.2](#))).

In [Chapter 9](#), we described the main resource for [cross-lingual supervision](#), i.e., [parallel corpora](#). For our experiments, we used the EUROPARL corpus, from which we selected a set of 10 languages ([9.1](#)). Prior to [compound identification](#), the [parallel corpus](#) has to be preprocessed ([9.2](#)). For our experiments, we overcome these preprocessing steps by exploiting an already preprocessed version of EUROPARL, provided by OPUS ([9.3](#)).

In [Chapter 10](#), we performed two pilot studies. The first study was the [Linguistic Criterion Inspection \(LCI\)](#) ([10.1](#)). Here, we created a gold standard of [nominal compounds](#) and ratings for [compoundhood](#) and for the validity of six [linguistic criteria](#) for [compoundhood](#), the [Europarl Nominal Compoundhood Ratings \(ENCR\)](#) ([10.1.1](#)). Based on the [ENCR](#), we performed various experiments for finding the [linguistic criteria](#) that correlate best with the [compoundhood status](#) ([10.1.3](#)). One result of the [LCI](#) was that the [spelling criterion](#) shows the highest [IAA](#) but the weakest correlation to the [compoundhood](#). Nevertheless, we considered this to be a language-dependent trend and expected a higher correlation between [spelling](#) and [compoundhood](#) for other languages.

Thus, we performed a second pilot study concerning [cross-lingual equivalents](#), i.e., the [Cross-lingual Compound Inspection \(XCI\)](#) (10.2).

In [Chapter 11](#), we explained the [cross-lingually supervised](#) method for [compound identification](#). In a preprocessed [parallel corpus](#), we first select [compound](#) candidates using a set of predefined [PoS patterns](#) (11.1). For each candidate, all available [cross-lingual equivalents](#) are retrieved (11.2). The final step is the [cross-lingual](#) validation, where [compound](#) candidates are accepted or discarded based on the amount of [compound equivalents](#) (11.3).

In [Chapter 12](#), we applied our [compound identification](#) method to the EUROPARL corpus, leading to the [Europarl Nominal Compound Database \(ENCD\)](#). We discussed some statistics and [cross-lingual](#) observations in the [ENCD](#) (12.1) and described some additional information provided with the [ENCD](#) (12.2).

In [Chapter 13](#), we showed some experiments for evaluating the performance of our [compound identification](#) method. Therefore, we considered all [ENCD](#) entries (for [CCR\(0\)](#) to [CCR\(4\)](#)) that occur in EUROPARL sentences inspected in the [ENCR](#) datasets (13.1). For each [ENCR](#) dataset, we provided results illustrating the potential of [cross-lingually supervised compound identification](#) (13.2).

Finally, in this [Chapter 14](#), we summarize Part [C](#) (14.1), answer the research questions posed in [Section 7.2](#) (14.2) and point to possible future work (14.3).

## 14.2. Conclusion

In this section, we answer the research questions posed in [Section 7.2](#).

**RQ\_1-A:** What [linguistic criteria](#) help to identify [compounds](#)?

⇒ In the experiments on the [LCI](#) (10.1.3), we observed that there are three [linguistic criteria](#) that correlate best with the [compoundhood](#) rating: the *inseparability*, the *inability to modify the modifier* and the *prosody*. As expected, the [criterion](#) that correlates least with [compoundhood](#) is the *spelling*, because English is an [open compounding language](#), i.e., most true [compounds](#) are realized in multiple [words](#).

**RQ\_1-A-i:** Which [linguistic criteria](#) show highest and lowest [IAA](#)?

⇒ The *spelling* [criterion](#) has the highest [IAA](#). This is to be expected, because judging whether a [target](#) expression is spelled as one or several [words](#) is straightforward. The [linguistic criterion](#) with the lowest [IAA](#) is *prosody*. This observation is in line



with previous work arguing that the prosody of **compounds** varies across speakers and dialects (Nakov, 2013).

**RQ\_1-A-ii:** What is the **identification** agreement, serving as UPPER bound for our **compound identification** method?

⇒ Although we used two native-speaking annotators who were introduced into the controversy of **compound** definition and iteratively trained on a common set of 20 EUROPARL sentences, the annotators developed their own notion of **compoundhood** during a stage of annotating EUROPARL sentences individually, leading to a final **identification** agreement which is only moderate. More specifically, they achieve an Jaccard coefficient of **0.431** and an F<sub>1</sub>-Score of **0.603** (Table 10.3), serving as UPPER bound for our **compound identification** method.

**RQ\_1-A-iii:** What is the agreement for rating **compoundhood**?

⇒ Other than the decision for **identifying** compoundlike expressions (Table 10.3, **RQ\_1-A-ii**), there is a solid agreement on the **compoundhood** rating of a commonly **identified** candidate (Table 10.6). More specifically, the average difference on the **compoundhood** rating (for 66 commonly **identified compounds**) is **0.227** (the possible rating differences range between 0 and 2).

**RQ\_1-B:** What are the most frequent formations of **cross-lingual equivalents** of an English **compound**?

⇒ In the **XCI**, we observed that for all nine **support languages** (Table 10.17), the most frequent formations of **cross-lingual equivalents** is a closed **compound** (CN), followed by a **complex nominal** (SN FC SN) and a simplex noun (SN). For Germanic **closed compounding languages** (Table 10.18), by a wide distance, the most frequent aligned **USP** is the closed **nominal compound** (CN, with 70.6%), followed by a simplex noun (SN) with 16.5%. This result of one-word **equivalents** in Germanic **support languages** was used for our **compound identification** method. For Greek (Table 10.19), the most frequent formation is a **2NC** (SN SN), whereas for Romance **support languages** (Table 10.20), the most frequent aligned **USP** is a **complex nominal** (SN FC SN) followed by an **NP** with a postnominal adjective (SN ADJ).

**RQ\_1-C:** Is **cross-lingual** information beneficial for the automatic **identification** of **compounds** in context?

⇒ The experiments outlined in Chapter 13 (i.e., Tables 13.1, 13.2 and 13.3) showed that the **Closed Compound Restrictor (CCR)** is a condition that can be used for **identifying compounds** with a high precision. The higher the degree of **closed compounding** among the **cross-lingual equivalents** ( $\Xi_{closed}$ ) of the **target compounds**, the higher the precision of correct **identification**. For the **Combination** dataset (containing annotations of both annotators, Table 13.3), the highest precision (0.839) is achieved with  $\Xi_{closed} = 4$ .

**RQ\_1-C-i:** What are the limitations of the use of **cross-lingual** evidence for **compound identification**?

The main limitation of **identification** based on the degree of **closed compounding** among all **cross-lingual equivalents** is **coverage**, i.e., there are a lot of true **compounds** that are translated into **phrasal equivalents**, by one or even more aligned **closed compounding languages**. The observation of phrasal translations was already discussed in Section 5.2. As a consequence, the **CCR** condition is too restrictive and thus leads to a strong recall and  $f_1$ -Score drop when increasing the parameter  $\Xi_{closed}$ . While the best recall is achieved in the **PoS pattern** baseline (**CCR(0)**) with 0.546, it drops down to 0.063 for **CCR(4)**.

### 14.3. Future Work

**Head category** In this thesis, we focus on the **identification** of the majority of **compounds**, viz. **nominal compounds**, i.e., **compounds** having a noun as **head**. We expect that our **cross-lingually supervised** approach is applicable to other categories as well. For example, the English **synthetic adjectival compound** *home made* is translated to German as *selbstgemacht* (`dict.cc`).

**LCI** In the **Linguistic Criterion Inspection (LCI)** (10.1), we explored the correlation between the validity of **linguistic criteria** and **compoundhood** based on human ratings. One limitation in the **LCI** is that we only considered expressions with a minimum **compoundhood** rating of 1. For a more representative correlation between **compoundhood** and the **linguistic criteria**, we also need **LC** ratings for expressions having a **compoundhood** rating of 0. We will add samples of not **identified word**

sequences (stipulating a **compoundhood** rating of 0) and let annotators rate the validity of the six **linguistic criteria**.

**XCI** In the **Cross-lingual Compound Inspection (XCI)** (10.2), we explored the spelling formation of **cross-lingual equivalents** of English **2NCs**, as given by an external gold standard provided by Ó Séaghdha (2007). For a more representative result, we would need **cross-lingual equivalents** of true bipartite non-**compounds** (e.g., phrasal adjective-noun sequences). To this end, we need resources of true non-**compounds** to be matched with EUROPARL. This additional step will be addressed in future work.

Moreover, we only considered **2NCs**. We expect to see a smaller degree of cross-lingual **closed compounding** for English **nominal compounds** with three or more **constituents**. To this end, we need resources of longer **nominal compounds** for which there is a sufficient overlap with EUROPARL. This extension to **compounds** with three or more **constituents** or to other **constituent** categories (e.g., adjective-noun **compounds**) will be addressed in future work.

**PoS patterns** In our experiments on the quality of the **ENCD** (and thereby on the performance of our proposed **compound identification** method), we observed that the **CCR(0)** baseline reaches a recall of only **0.546** (for the **ENCR Combination** dataset, Table 13.3). The main reason for this is that most false negatives are represented by **PoS** sequences including a named entity. These expressions are not covered by our predefined **PoS patterns**. We will enrich the **compound identification** method with this missing **PoS pattern** in future work.

**CCR** Evaluating the performance of our proposed **identifier**, we observed that the **CCR** condition is too restrictive leading to a strong recall drop when increasing the parameter  $\Xi_{closed}$ . In future, we will try to mitigate this recall drop by adding alternative conditions based on further **linguistic criteria**.

## *14. Bottom Line of Compound Identification*

Part D.

Compound Splitting



# 15. Introduction to Compound Splitting

In this part, we present and elaborate the work published in Ziering and Van der Plas (2016), Ziering et al. (2016) and Jagfeld et al. (2017).

While there are various languages that mainly create **open compounds**, i.e., **open compounding languages** such as *English*, or languages that hardly perform **compounding** at all such as Romance languages (e.g., *French* or *Italian*), that use **complex nominals** instead, there are many languages that create **closed compounds**, i.e., **closed compounding languages** such as most Germanic languages (e.g., German, Swedish or Dutch).

In order to get to the **understanding** of **closed compounds**, we need to analyze their structure, i.e., we need to determine their mediate and **immediate constituents**. For this purpose, previous work addressed the task of **compound splitting** (also called **decompounding**), i.e., transforming **closed compounds** to **open compound** equivalents. For example, the German **closed compound** *Hühnersuppe* ‘chicken soup’ is **split** into the **constituent forms** *Hühner* and *suppe* and into the **constituent lemmas** *Huhn* ‘chicken’ and *Suppe* ‘soup’. This **normalization** to **constituent lemmas** is necessary, because downstream applications such as **Statistical Machine Translation (SMT)** systems expect **lemmas** as input. **Compound splitting** is an essential component for many **NLP** tasks such as **SMT**, **Information Retrieval (IR)**, **Speech Recognition (SR)** or **Recognizing Textual Entailment (RTE)**.

## 15.1. Motivation

### 15.1.1. The Common Statistical Approach

Most previous work on statistical **compound splitting** follows a two-step **generate-and-rank** procedure (as will be discussed in Section 16.1).

## Split Generation

Firstly, all possible or morphologically plausible candidate splits are generated. This can be done by enumerating all possible [split points](#) or by collecting all morphological analyses from an external tool (e.g., SMOR). For the [constituent forms](#) resulting from a candidate split, the underlying [constituent lemmas](#) are determined by [normalizing](#) the potential forms with a **hand-crafted** set of morphological rules (e.g., truncating [linking elements](#) such as the  $\oplus$ -suffix).

## Ranking

In the second step, all candidate splits are scored according to a set of statistical features, such as the geometric mean of the corpus frequencies of the most plausible [constituent lemmas](#) (Koehn and Knight, 2003). The top-ranked candidate [split](#) is selected as [splitting](#) decision.

## Evaluation

Previous work on [compound splitting](#) presents two ways of evaluating the performance.

Firstly, [splitting](#) quality is evaluated [extrinsically](#) by incorporating the [splitter](#) in an [SMT](#) system. A sentence from a [closed compounding](#) (source) language (e.g., German) is translated to an [open compounding](#) (target) language (e.g., English). [Compound splitting](#) prior to translation improves the quality of the translated sentence, in particular if the [closed compound](#) is unknown to the trained [SMT](#) system. The better the translation quality, the better the [compound splitter](#) used a priori.

Secondly, [compound splitters](#) are evaluated [intrinsically](#) either by inspecting the correctness of the predicted [split points](#) or by string matching the gold [constituent lemmas](#) with the predicted [constituent lemmas](#).

### 15.1.2. Limitations of the Common Statistical Approach

#### Language-specific Limitations

While a corpus lookup for a hypothesized [constituent lemma](#) can be considered as a language-independent step, the prior [normalization](#) of potential [constituent forms](#) to these [constituent lemmas](#) is non-trivial and usually requires language-specific knowledge about the morphological nature of [closed compounds](#) (e.g., about [linking elements](#)), some of which is discussed in Section 3.9. As a consequence, most previous work on [compound](#)



**splitting** includes language-specific knowledge such as large lexicons and morphological analyzers (Fritzinger and Fraser, 2010) or hand-crafted lists of **linking elements** and rules for modeling morphological transitions of **constituent inflection** (Koehn and Knight, 2003, Stymne, 2008, Weller and Heid, 2012).

This makes the approaches **language-dependent** and non-applicable to foreign languages without the effort of manually adding morphological information to the system.

## Limitations of using Corpus Frequency

By considering each **constituent** in isolation, approaches limited to **constituent** frequency neglect the semantic compatibility between a **compound** and its **constituents**. For example, while *Eidotter* ‘egg yolk’ has the intended meaning of the yolk of an egg (i.e., *Ei|dotter*), the low frequency of *Dotter* ‘yolk’ often makes frequency-based **splitters** rank a less plausible interpretation higher: *Eid|otter* ‘oath otter’.

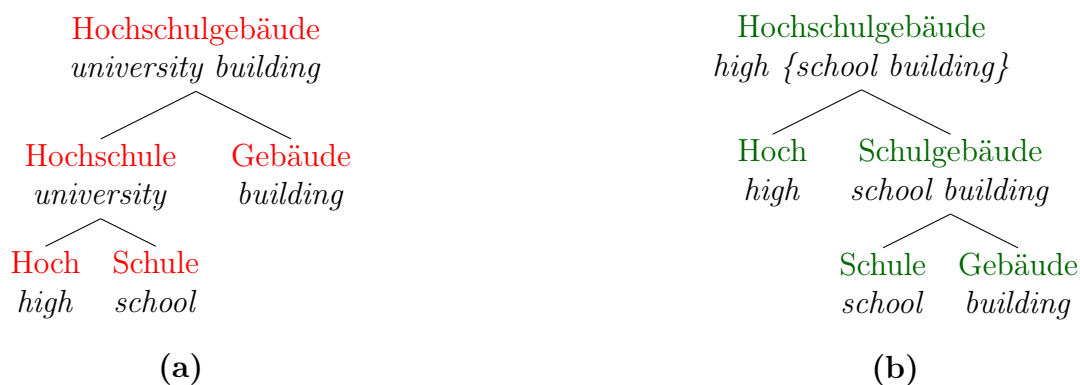


Figure 15.1.: Structures for the German *Hochschulgebäude* ‘university building’

Besides selecting the false **constituent lemmas**, another source of trouble is the **bracketing** of  $N$ -partite ( $N > 2$ ) **compounds**, such as the German **three-Noun Compound (3NC)** *Hochschulgebäude* ‘university building’ (lit.: *high school building*). Figure 15.1 shows the two possible structures with the related meaning. While the correct LEFT-branched structure, which means a building for the university (or high school), is shown in Figure 15.1(a), the high corpus frequency of *hoch* ‘high’ promotes the false RIGHT-branched structure, shown in Figure 15.1(b), which would mean a high building for a school.

## Limitations of Existing Evaluation Methods

### Intrinsic Evaluation

The common [intrinsic evaluation](#), outlined in Section [15.1.1](#) has some limitations.

**Evaluation of Constituent Lemmas.** Using exact string match between the gold [constituent lemma](#) and the predicted [constituent lemma](#) is too restrictive. It disregards cases where there are different variants of a [lemma](#) (e.g., due to different spelling conventions) or where several [constituent lemmas](#) are plausible for a given [constituent form](#). For example, for the German [compound](#) *Tanzlokal* ‘dance hall’ [split](#) into *Tanz* | *lokal*, the [modifier](#) can be both the verb *tanzen* ‘to dance’ or the converted noun *Tanz* ‘dance’ without changing the meaning of the [compound](#). Actually, DICT.CC provides two English translations for the German *Tanzlokal*: ‘dance hall’ and ‘dancing hall’.

**Evaluation of Split Points only.** Evaluating only the [split point](#) selection is not a solution for this issue, because [constituent normalization](#) is a non-trivial challenge which should not be delegated to a downstream [lemmatization](#) step, because there are non-paradigmatic [constituent forms](#), a state-of-the-art [lemmatizer](#) is not trained for, (e.g., the German *Armut-s* ‘poverty’ as in *Armutsbekämpfung* ‘poverty elimination’). Moreover, there are cases of [word-sense ambiguity](#) for [constituent forms](#) which would be resolved when determining the underlying [constituent lemma](#) (e.g., the [constituent form](#) *Streich* in the German [compound](#) *Streich|käse* ‘spread cheese’ can be [normalized](#) to the noun *Streich* ‘prank’ or to the verb *streichen* ‘to spread’).

**Dual Evaluation.** Previous work that exclusively evaluates the predicted [split points](#) without [normalization](#) (e.g., Riedl and Biemann (2016)) cannot be compared to previous work that exclusively evaluates the predicted [constituent lemmas](#) (e.g., Koehn and Knight (2003)). Therefore, we argue to provide evaluations for both the correct [split points](#) (or [constituent forms](#)) and the correct [constituent lemmas](#).

### Extrinsic Evaluation

As discussed in Section [15.1.1](#), the most widely used external task for the [extrinsic evaluation](#) of [compound splitting](#) is [SMT](#). However, there are some issues with [SMT](#) which make it less suitable for the goal of [extrinsic evaluation](#) of [compound splitting](#).

In to-English SMT methods, **closed compounds** which are listed in the translation dictionary do not need to be **split** for getting translated correctly. This means that **undersplitting** is not penalized consistently.

Moreover, as discussed by Dyer (2009), **oversplitting** might be ignored, because **words** that are oversplit would be learned as phrases in a phrase-based SMT system.

A **constituent**-wise translation also fails for **compounds** having a non-literal translation such as *Krankenversicherung* ‘health insurance’ (lit: ‘invalid insurance’) (see asymmetric translations in Section 5.3).

The “*general notion of quality of a translation is [...] subjective*” (Olive et al., 2011, sec. 5.1.2) and there are many possible translations for a source expression. For the German example *Tanzlokal*, discussed above, there are two possible translation to English: ‘dance hall’ and ‘dancing hall’.

Finally, common SMT methods (e.g., the MOSES toolkit (Koehn et al., 2007)) have an expensive runtime complexity, which hinders a flexible and efficient development of **compound splitting** methods.

## 15.2. Contributions and related Research Questions

In this thesis, we address all limitations of common **compound splitting** approaches, outlined in Section 15.1.2, and try to overcome these with the following **three methodological contributions**. Moreover, in this section, we repeat and refine some research questions posed in Section 1.3 and add some additional ones.

### 15.2.1. Multilingual Compound Splitter

We develop a **multilingual compound splitter**, for which there is no need for language-specific knowledge about **constituent inflection**. Macherey et al. (2011) were the first to overcome the limitation of language specificity by learning morphological **compounding** operations automatically from **parallel corpora**. We would like to adopt this idea but take it one step further by avoiding the usage of **parallel data**, which are known to be sparse and frequently domain-specific, while Bretschneider and Zillner (2015) showed that **compounding** morphology varies between different domains. Instead, we exploit **lemmatized** corpora and use regular **word inflection** as an approximation to **constituent inflection**. The morphological operations are modeled using **Morphological Operation Patterns** (MOPs), outlined in Chapter 17. This way, we are able to process **compounds**

of any type of domain.

**RQ\_2-A:** What sources of [indirect supervision](#) can we use for [compound splitting](#)?

**RQ\_2-A-i:** How well does the approximation of using [word inflection](#) for [constituent inflection](#) work for [compound splitting](#)?

**RQ\_2-A-ii:** How expressive are the proposed [MOPs](#)?

- Are there morphological operations that cannot be modeled?
- How ambiguous are these patterns and how is ambiguity resolved?

**RQ\_2-B:** How do manual-resource-lean methods compare to resource-rich and language-specific approaches?

**RQ\_2-B-i:** What is the difference in [splitting](#) performance when working with operations for [word inflection](#) instead of [constituent inflection](#)?

**RQ\_2-B-ii:** How competitive is the [multilingual splitting](#) approach compared to language-specific [splitting](#) methods?

**RQ\_2-C:** How language-independent are our [splitting](#) approaches and what resources do they still need?

### 15.2.2. Semantics-driven Re-ranker for Compound Splitters

We propose a re-ranker for frequency-based [compound splitters](#) based on [Distributional Similarity](#) ([Dsim](#)) between the intended meaning of [target compound](#) and its potential [constituents](#). This way, we address the limitations outlined in Section 15.1.2: we combine information about semantic plausibility with corpus frequency and thereby mitigate the impact of high frequent [constituents](#), often leading to implausible [compound analyses](#). For example, knowing that *Eidotter* is distributionally more similar to *Dotter* than to *Otter* promotes the correct [compound split](#) (i.e., *Ei|dotter* ‘egg yolk’).

**RQ\_2-D:** How effective is the [Dsim](#) information for [compound splitting](#)?

**RQ\_2-D-i:** What is the average performance gain when adding [Dsim](#) information?

**RQ\_2-D-ii:** Which frequency-based [compound splitter](#) benefits most from adding semantics information?

- How can linguistically-informed [splitting](#) systems be improved when adding semantics information?
- How does distributional similarity improve statistical [compound splitters](#)?

**RQ\_2-D-iii:** What are the individual contributions of [Dsim](#) and corpus frequency information?

**RQ\_2-D-iv:** What [constituent type](#) provides the best-working [Dsim](#) information?

- In which cases does the [modifier](#) outperform the [head](#)?
- In which cases does the [head](#) outperform the [modifier](#)?
- Which type of combination of [modifier](#) and [head](#) work best?

### 15.2.3. Additional Evaluation Methods

As discussed in Section 15.1.2, previous work performs an [intrinsic evaluation](#) either on the [constituent forms](#) or on the [constituent lemmas](#), leading to several limitations. Moreover, most previous [compound splitters](#) were evaluated [extrinsically](#) on the task of [SMT](#).

We present two novel ways of evaluating [compound splitting](#). Firstly, we propose a novel [intrinsic evaluation](#) method by treating [split point](#) detection and [constituent normalization](#) as two separate disciplines. This way, we are able to compare to both previous work evaluating on [constituent forms](#) and previous work evaluating the predicted [constituent lemmas](#). Moreover, we have a more fine-grained perspective on the quality of a [compound splitter](#), e.g., a method which succeeds to determine the correct [split point](#) but misses the required gold [constituent lemmas](#) is better than a system that fails in both [split point](#) detection and [constituent normalization](#). The novel [intrinsic evaluation](#) method is presented in Section 18.6.5.

In addition, we propose a novel [extrinsic evaluation](#) method for [compound splitting](#) as an alternative for the commonly used task of [SMT](#). This [extrinsic evaluation](#) method makes use of a language-independent [Recognizing Textual Entailment \(RTE\)](#) system. This approach is presented in Chapter 20.

**RQ\_2-E:** How suitable are the novel [intrinsic](#) and [extrinsic evaluation](#) methods for [compound splitting](#)?

**RQ\_2-E-i:** Are there differences in the ranking of [compound splitters](#) for [split point](#) determination and [constituent normalization](#)?

- What methods perform best with respect to [split point](#) determination?
- What systems are superior in [constituent normalization](#)?

**RQ\_2-E-ii:** How does [RTE](#) treat the different errors occurring in [compound splitting](#): [false splitting](#), [oversplitting](#) and [undersplitting](#)?

### 15.3. Outline

The thesis part [D](#) is structured as follows.

**Chapter 16** gives an overview about **previous work** on [compound splitting](#).

**Chapter 17** presents the concept of a **Morphological Operation Pattern (MOP)**, that is used for learning and applying a morphological transformation for [compound splitting](#), i.e., [constituent inflection](#).

In **Chapter 18**, a **multilingual compound splitting method** is described.

While this [splitting](#) method is mainly driven by corpus frequency of the assumed [constituent lemmas](#) and the exploited morphological patterns, in **Chapter 19**, we aim to overcome the limitations of a purely frequency-based approaches (outlined in Section [15.1.2](#)) with a flexible way of enriching a [compound splitting](#) model with **Distributional Similarity (Dsim)**.

While previous work on [compound splitting](#) mainly used [SMT](#) as [extrinsic evaluation](#), in **Chapter 20**, we propose a promising alternative [NLP](#) task for the [extrinsic evaluation](#) of [compound splitting](#): **RTE**.

The [compound splitting](#) part [D](#) is **summarized and concluded** in **Chapter 21**. Finally, we give an outlook on **future work** concerning all three contributions.

# 16. Related Work on Compound Splitting

In this chapter, we outline previous related work on [compound splitting](#). While discussing all publications about [compound splitting](#) would exceed the scope of this thesis, we focus on the most important and influential approaches, which are most relevant for the contributions claimed in this thesis (see Section [15.2](#)).

In the description of each approach, we focus on four features:

1. **Splitting approach** - Although previous work on [compound splitting](#) cannot be grouped into clear categories, since most approaches are kind of hybrid, there are two main lines of [compound splitting](#) approaches often discussed in literature: **statistical** (or corpus-based) and **linguistic** approaches. Moreover, this feature describes **all information** that is used for the [splitting](#) task.

The [compound splitting](#) method presented in this thesis is designed in the statistical spirit of Koehn and Knight (2003), i.e., it ranks all possible binary splits according to the geometric mean of the potential [constituents](#)' scores. Furthermore, we are enriching a [compound splitting](#) method with [Distributional Similarity \(Dsim\)](#) information for promoting more plausible splits.

2. **Constituent inflection** - An important aspect of a [compound splitting](#) approach is the way it deals with [constituent inflection](#). Most [compound splitters](#) aim to determine the composed [lexemes](#) (rather than only the [constituent forms](#)). Due to [constituent inflection](#), the [constituent forms](#) have to be [normalized](#) using a morphological transformation rule (e.g., the truncation of the [linking element](#) *s*, as in the German [compound](#) *Kindheits|erinnerung* ‘childhood memory’). These rules differ from language to language. While there are linguistic approaches (Section [16.2](#)) based on morphological analyzers (e.g., SMOR), most [compound splitters](#) discussed in this chapter are working with a small hand-crafted or automatically compiled list of morphological operations modeling [constituent inflection](#). In our

discussion about this feature, we provide information about whether morphological knowledge is added manually or whether it is learned automatically. Moreover, we describe the different knowledge resources about **constituent inflection** (e.g., different sets of **linking elements**). Langer (1998) conducted a corpus study about German **compounds** and presented a list of the 20 most frequent morphological operations (which a **modifier** undergoes when being involved in a **compounding** process, i.e., **constituent inflection**). This collection<sup>1</sup> is utilized by various German **compound splitting** methods.

Similar to Macherey et al. (2011), our goal is to avoid a hand-crafted language-specific list of morphological transformations modeling **constituent inflection**. Therefore, we try to approximate **constituent inflection** with **word inflection**, which works for many - in particular Germanic - languages (Ziering and Van der Plas, 2016).

3. **Target languages** - Most previous work on **compound splitting** focuses on **closed compounding** and morphologically rich **target languages**, i.e., languages that produces exclusively **closed compounds** such as the Germanic languages (e.g., *German* or *Dutch*). *English* as an **open compounding language** has only a few **closed compounds**; and those are mostly created by **lemma concatenation** (i.e., without any **constituent inflection**, e.g., *dog|house*). Thus, **compound splitting** is less researched for *English* and other **open compounding languages**. Most previously developed **compound splitters** are designed for one language, i.e., there are very few **multilingual compound splitters** such as the one developed by Macherey et al. (2011).

The **compound splitting** method presented in this thesis is designed for being applicable to many **target languages**. The method described in Chapter 17 and Chapter 18 avoids the use of any language-specific information about **constituent inflection**. This **compound splitting** approach is tested for the **target languages** *German*, *Dutch* and *Afrikaans*. Analogously, the enrichment of a **compound splitter** using **Distributional Semantics (DS)**, described in Chapter 19, is also designed language-independently. The impact of adding **DS** information to **compound splitting** is tested for *German*. During the **extrinsic evaluation** outlined in Chapter 20, we work on a language-independent **Recognizing Textual Entailment (RTE)** framework and entailment algorithm. Restricted by the availability of **RTE** test data for **closed compounding languages**, we test three German **compound splitters** on **RTE**.

---

<sup>1</sup>Appendix B



4. **Evaluation** - The most influential [compound splitting](#) publication in previous work, Koehn and Knight (2003), proposed a benchmarking method for the [intrinsic evaluation](#) of [compound splitting](#) using the common measures: precision, recall, F<sub>1</sub>-Score and accuracy. An alternative evaluation method (the [extrinsic evaluation](#)) is to apply [compound splitting](#) to the input data of an external downstream NLP task which benefits from [split compounds](#), and to use the [intrinsic evaluation](#) method of the external task. While there are many task that can benefit from [compound splitting](#) (e.g., RTE, as presented in Chapter 20), previous work based on an [extrinsic evaluation](#) method mostly focuses on the task of SMT, i.e., comparing the performance of SMT with and without prior [compound splitting](#) using state-of-the-art metrics for SMT, such as BLEU score (Papineni et al., 2002).

In this thesis, [compound splitting](#) is evaluated both [intrinsically](#) and [extrinsically](#). In contrast to previous work, the [intrinsic evaluation](#) presented in this thesis measures the correctness of both disciplines: (1) determining the correct [split points](#) (or [constituent forms](#)) and (2) [normalization](#) to the correct [constituent lemmas](#). In the [extrinsic evaluation](#), we are presenting a novel external task: RTE. We show that RTE can be improved using [compound splitting](#) information, and differences in RTE performance (when being enriched with [splitting](#) information provided by different methods) can reveal differences in the quality of the [splitting](#) approaches.

For structuring this chapter, we group previous work with respect to the [compound splitting](#) approach.

## 16.1. Statistical Approaches

[Compound splitting](#) approaches which are mainly based on corpus statistics (e.g., frequency, entropy, [Mutual Information](#) (MI), ...) have the benefit of being flexible, i.e., they can easily be adapted to other domains and languages without the need of much specific information.

### 16.1.1. Frequency-based Approaches

For a given target [word](#), Larson et al. (2000) counted the number of corpus [words](#) starting and ending with any prefix and suffix. In the next step, for each character transition, the difference of prefix and suffix counts is determined. The potential [split points](#) are the local maxima of both prefix and suffix, as illustrated in an example

## 16. Related Work on Compound Splitting

provided by Larson et al. (2000), given in Table 16.1. The **compound** *Friedenspolitik* ‘policy of peace’ is **split** at the joint local maxima into *Friedens* ‘peace’ and *politik* ‘policy’. Larson et al. (2000) focused on *German* as **target language**.

Target word		F	r	i	e	d	e	n	s	p	o	l	i	t	i	k
counts	prefix	-	-	39	29	29	25	24	23	3	1	1	1	1	1	1
	suffix	1	1	1	1	1	2	7	37	88	89	89	92	99	-	-
$\Delta$ counts	prefix	-	-	-	10	0	4	1	1	20	2	0	0	0	0	0
	suffix	0	0	0	0	0	5	30	51	1	0	3	7	-	-	-
max $\Delta$ counts	prefix	-	-	-			*			*						
	suffix								*			-	-	-	-	

Table 16.1.: Splitting example from Larson et al. (2000)

Besides **SMT**, an alternative downstream task for the **extrinsic evaluation** of **compound splitting** is **Speech Recognition (SR)**, where **compounds** occurring in the recognition lexicon are **split** into linguistically meaningful sub-units, in order to reduce the **Out-Of-Vocabulary (OOV)** rates (Larson et al., 2000) or to improve the letter-to-sound conversion (Adda-decker et al., 2000). For example, *Eidotter* can have two possible analyses: *Ei|dotter* ‘egg yolk’ and *Eid|otter* ‘oath otter’. While *Ei|dotter* has the phonetic transcription [ai|dɔtɐ], *Eid|otter* is pronounced as [ait|ɔtɐ] (Cap, 2014). Larson et al. (2000) evaluated their **compound splitter extrinsically** on the task of **SR**.

**Monz and de Rijke (2001)** iterated over all characters of a target noun (from left to right) and tried to match the resulting prefixes with a lexicon. In the case of a match, the remaining suffix is recursively processed, leading to a right-branching **split tree** structure. Monz and de Rijke (2001) allowed for an  $\oplus$ s suffix, whereas Koehn and Knight (2003), as will be discussed shortly, used two fillers for **normalization**:  $\oplus$ s and  $\oplus$ es. The focused **target languages** of Monz and de Rijke (2001) were *German* and *Dutch*. They performed an **intrinsic evaluation** on the predicted **constituent lemmas** and an **extrinsic evaluation** based on **IR**.

The most influential **compound splitting** publication in previous work is **Koehn and Knight (2003)**. They proposed to use corpus frequency of the potential **constituents** for ranking all possible splits with a **constituent** length of at least 3 characters, including the **non-split option**, i.e., an **atomic** analysis for non-compounds. The splits are ranked according to the geometric mean of the frequencies. Koehn and Knight (2003) performed an **intrinsic evaluation** on the predicted **constituent lemmas** and focused on *German* as **target language**. Although high-frequent **closed compounds** can lead to **undersplitting**

(as shown in Table 16.2 for the compound *Aktionsplan* ‘action plan’), the method of Koehn and Knight (2003) has often been adopted in later approaches.

Constituents	Frequencies	Geometric mean score
<i>Aktionsplan</i> ‘action plan’	852	852
<i>Aktion</i> ‘action’ • <i>Plan</i> ‘plan’	960 • 710	825.6
<i>Aktions</i> ‘action (s-suffix)’ • <i>Plan</i> ‘plan’	5 • 710	59.6
<i>Akt</i> ‘act’ • <i>Ion</i> ‘ion’ • <i>Plan</i> ‘plan’	224 • 1 • 710	54.2

Table 16.2.: Example of **undersplitting** in Koehn and Knight (2003)

For example, it has been adopted by **Stymne (2008)**, who performed several experiments to measure the impact of varying parameters of a modified version of the algorithm of Koehn and Knight (2003) for factored **SMT**. She observed that **splitting** parameters should not necessarily be the same for the translation in different directions. The modifications and varying parameters she introduced to the algorithm of Koehn and Knight (2003) include:

Mean scorer (arithmetic vs. geometric)
Length of valid <b>constituents</b> and <b>compounds</b>
Maximum number of <b>constituents</b>
Valid <b>compound class</b> ( <b>content words</b> vs. all words)
Optional <b>PoS</b> equality between <b>head</b> and <b>compound</b>
Additional morphological operations (e.g., those proposed by Langer (1998))
Corpus frequencies of <b>constituent forms</b> and/or <b>constituent lemmas</b>
Hyphens as exclusive <b>split points</b> for <b>hyphenated compounds</b>

Table 16.3.: Modifications proposed by Stymne (2008)

While Stymne (2008) re-implemented the approach of Koehn and Knight (2003), she used the collection of Langer (1998) for modeling **constituent inflection**. As **target language**, Stymne (2008) focused on *German*. She performed an **intrinsic evaluation** on the predicted **constituent lemmas**, as well as an **extrinsic evaluation** on the task of **SMT**.

In a similar way, **Stymne and Holmqvist (2008)** exploited the empirical approach for **compound splitting** in Swedish-English **SMT**. They used the arithmetic mean score, as suggested by Stymne (2008). Swedish **compounds** undergo a special spelling transformation: if two **constituents** are to be joint such that there would be three equal consonants in a row, one such consonant is dropped (Stymne and Holmqvist, 2008). For modelling this behaviour, a third consonant is allowed if a **split point** separates two

## 16. Related Work on Compound Splitting

consecutive consonants (e.g., *stop|plikt*  $\rightarrow$  *stopp plikt* ‘stop obligation’). Besides the operations proposed by Langer (1998), Stymne and Holmqvist (2008) used two additive suffixes  $\oplus$ s and  $\oplus$ t, two subtractive suffixes  $\ominus$ e and  $\ominus$ a, as well as nine combinations of suffixation and truncation (e.g.,  $\ominus$ a/ $\oplus$ s). Stymne and Holmqvist (2008) focused on *Swedish* as [target language](#) and performed an [intrinsic evaluation](#) on the predicted [constituent lemmas](#), as well as an [extrinsic evaluation](#) on the task of SMT.

Other previous work relying on an adaptation of the splitting method of Koehn and Knight (2003) for improving SMT include [Popović and Ney \(2004\)](#), [Yang and Kirchhoff \(2006\)](#) and [Durgar El-Kahlout and Yvon \(2010\)](#).

For the unsupervised [multilingual](#) and [cross-lingual compound splitter](#) developed by [Macherey et al. \(2011\)](#), the authors used a dynamic programming approach based on Bayes’ decision rule and corpus frequency of the potential [constituents](#). Macherey et al. (2011) defined a cost function that includes information about which morphological operations are applied at which position of which [constituent](#). Macherey et al. (2011) were the first to overcome the need for manual morphological input and the limitation to a fixed set of [linking elements](#) by learning morphological operations automatically from [parallel corpora](#) including a [support language](#) which creates [open compounds](#) and has only little [word inflection](#), such as *English*. For example, the German [compound](#) *Überweisungsbetrag* is translated to *English* as *transfer amount*. The back-translations of the English [constituents](#) yield to *Überweisung* and *Betrag*. Using the Levenshtein [Edit Distance \(ED\)](#) algorithm between the [target compound](#) and the concatenation of the back-translations yield the [linking element](#):  $\oplus$ s. Macherey et al. (2011) did not focus on a specific [target language](#). They observed an improvement in SMT performance for various [target languages](#) in different language families. They tested their [multilingual](#) method on *German, Danish, Norwegian, Swedish, Greek, Estonian* and *Finnish* using SMT as [extrinsic evaluation](#) method. We take the approach of Macherey et al. (2011) one step further by switching the dependence on [parallel data](#), known to be sparse, to monolingual corpus [lemmatization](#). Moreover, while Macherey et al. (2011) weights each true morphological operation equally, our approach assigns different weights to the various [word inflection](#) operations, which improves the [split](#) disambiguation quality.

The goal of [Clouet and Daille \(2014\)](#) was to create a [multilingual compound splitting](#) method which can be used both purely corpus-based (applicable to different [target languages](#)) and with the support of manual linguistic knowledge. They generated all possible binary splits with a minimum [word](#) length of 3 characters (Koehn and Knight, 2003). The resulting [constituent forms](#) can be normalized using hand-crafted rules or

with string similarity (e.g., the [normalized](#) Levenshtein distance (Frunza and Inkpen, 2009)) to [lemmas](#) within a monolingual lexicon including [PoS-filtered lemmas](#) and neo-classical stems or within a corpus. The [constituents](#) are scored using linear interpolation, where the parameters are learned with a small training set. The [splitting](#) and scoring method can be applied recursively on the [constituents](#) up to a predefined [splitting](#) depth. For [splitting](#) Russian [compounds](#), Clouet and Daille (2014) manually defined 15 ‘linguistic rules’ for the [constituent inflection](#) of the [modifier](#) and 14 rules for the [head](#). Clouet and Daille (2014) tested their [multilingual](#) method on the ‘non-prototypical’ (as they call them) [target languages](#) *English* and *Russian*, i.e., [target languages](#) which are infrequently addressed in previous [splitting](#) approaches. They performed an [intrinsic evaluation](#) on the predicted [constituent lemmas](#).

For the medical domain, **Bretschneider and Zillner (2015)** enumerated all possible splits as proposed by Koehn and Knight (2003). They disambiguated candidate splits using [semantic relations](#) from the medical domain ontology (e.g., *Beckenbodenmuskel* ‘pelvic floor muscle’ is binary [split](#) into *Beckenboden* | *muskel* using the [part\\_of](#) relation). As back-off strategy, if the ontology lookup fails, they used [constituent](#) frequency (Koehn and Knight, 2003). Bretschneider and Zillner (2015) compared the [splitting](#) performance between the two fillers of Koehn and Knight (2003) and the collection of morphological operations developed by Langer (1998), illustrating the necessity of an exhaustive set of [linking elements](#). Moreover, they showed that the data of Langer (1998) is still not sufficient for domain-specific [targets](#). They proposed seven further morphological transformations (e.g.,  $\oplus$ *ial*) observed in the medical language, a derivative of the Latin and Greek language (Bretschneider and Zillner, 2015). As [target language](#), Bretschneider and Zillner (2015) focused on *German*. They evaluated their method [intrinsically](#) based on the predicted [constituent lemmas](#). In Chapter 19, we will discuss our re-ranking method based on lexical semantics. In contrast to Bretschneider and Zillner (2015), we do not restrict to a certain domain and related ontology but use [DS](#) in combination with frequency-based [split](#) features for the disambiguation.

### 16.1.2. Approaches based on Cross-lingual Information

While the creation of a [parallel corpus](#) can be considered linguistically motivated (i.e., they are compiled manually using language experts), the exploitation of [parallel data](#) for [cross-lingual](#) approaches (e.g., by using a statistical [word aligner](#) such as MGIZA++ (Gao and Vogel, 2008)) can be considered as an unsupervised, statistical method.

**Brown (2002)** proposed an approach relying on a German-English [parallel corpus](#).

His approach is based on the observation that, particularly in the medical domain, there are cognate words between *German* and *English*, i.e., words being derived from the same etymological origin and thus having a high string similarity, such as *Herztransplantation* ‘heart transplantation’. A manually compiled set of cross-lingually corresponding characters (e.g., a German *i* often corresponds to an English *y*) helps to find an English word pair having the greatest string similarity<sup>2</sup> to a German target compound. The correct split point separates the target compound into constituents having the greatest string similarity to the determined English counterparts. In addition, Brown (2002) used a bilingual dictionary (that can be compiled from the parallel corpus) as back-off for cases, where there is no cognate relation. Brown (2002) focused on *German* as target language. The splitting method is extrinsically evaluated on the task of SMT.

In addition to the frequency-based approach, Koehn and Knight (2003) compiled a bilingual dictionary from parallel data and ruled out German compound splits having no one-to-one correspondence in English. For example, while *Aktionsplan* has a literal translation to ‘action plan’, the weekday name *Freitag* is not (literally) translated to a compound (i.e., ‘free day’) but to the atomic word ‘Friday’.

While not using cross-lingual information for the splitting procedure, Macherey et al. (2011) exploit bilingual resources including *English* and a closed compounding language (e.g., *German*) for learning constituent inflection operations, which are used in a frequency-based splitting approach, as discussed in Section 16.1.1.

### 16.1.3. Approaches based on Distributional Semantics

While semantics in general can be considered as a linguistic concept, Distributional Semantics (DS) relies on the statistical distribution of words in a corpus, and thus can be regarded as a statistical information type.

Daiber et al. (2015) developed a compound splitter relying on semantic analogy (e.g., *bookshop* is to *shop* as *bookshelf* is to *shelf*). From word embeddings of compound and head word, they learned prototypical vectors representing the modification. During splitting, they determined the most suitable modifier by comparing the analogy to the prototypes. Daiber et al. (2015) restricted to the two linking elements proposed by Koehn and Knight (2003). As target language, they focused on *German*. They evaluated their compound splitter intrinsically based on the predicted constituent lemmas and extrinsically on the task of SMT. In Chapter 19, we will discuss our re-ranking method

---

<sup>2</sup>Brown (2002) used the least common substring (LCS) as string similarity measure.

based on DS. While Daiber et al. (2015) developed an autonomous DS-based splitter and focused on semantic analogy, we developed a re-ranker that combines information about **Distributional Similarity** (Dsim) with additional **splitting** features (such as **constituent frequency**).

In contrast to Daiber et al. (2015), who generate all possible splits according to Koehn and Knight (2003), DS can also serve as indicator for the possible **split** candidates, as has been done by **Riedl and Biemann (2016)**. They deploy a pre-compiled **Distributional Thesaurus** (DT) for **identifying** all possible **constituents**: the corpus **tokens** that are distributionally most similar to the **compound** and constitute substrings of it. For ranking **split** options, they adapted the geometric mean scorer proposed by Koehn and Knight (2003): instead of pure frequency, the probability is used. Riedl and Biemann (2016) neglect the **normalization** of **constituent forms**. Assuming that **modifier** forms are frequently paradigmatic, they built a **DSM** on corpus **word forms**. Non-paradigmatic **constituent forms** (e.g., *Armut's*- ‘poverty’) are handled using a smoothing factor  $\epsilon$  (Riedl and Biemann, 2016, sec. 3.2). They tested their **multilingual** approach on *German* and *Dutch*, and performed an **intrinsic evaluation** on the predicted **constituent forms**. While the stand-alone method of Riedl and Biemann (2016) focuses on knowledge-lean **split point** determination, our approach improves any **compound splitter** including the task of **constituent normalization**.

#### 16.1.4. Approaches based on Supervised Machine Learning

**Marek (2006)** used a **weighted Finite State Transducer** (wFST) for **splitting** and **bracketing compounds**, based on a hand-crafted training corpus<sup>3</sup>. The wFST is automatically created using the AT&T FSM Library<sup>4</sup> developed by Mehryar Mohri, Fernando C. N. Pereira and Michael D. Riley. The weights are learned from the training corpus using 10-fold cross-validation. Marek (2006) used the collection from Langer (1998) and manually added a few further **linking elements** (Marek, 2006, p. 17). His method is **intrinsically** evaluated based on the predicted **constituent lemmas**.

**Alfonseca et al. (2008)** aimed to improve German **compound splitting** for noisy query keywords in **Information Retrieval** (IR). They trained a **Support Vector Machine** (SVM) using various features including booleans whether to **split** or not, **constituent frequency**, probability, entropy or **MI** of the **split**. For **splitting** a **target compound**  $w$ , the feature for  $w$  and all possible binary splits of  $w$  are generated. The classification

<sup>3</sup>This dataset will be used as a **compound splitting** gold standard in Chapter 18.

<sup>4</sup>[https://en.wikipedia.org/wiki/AT%26T\\_FSM\\_Library](https://en.wikipedia.org/wiki/AT%26T_FSM_Library)

output with the highest confidence is used as [splitting](#) decision (Alfonseca et al., 2008). Alfonseca et al. (2008) restricted to a selection of morphological operations from the collections of Langer (1998) and of Marek (2006). The [splitting](#) method was [intrinsically](#) evaluated based on the predicted [constituent lemmas](#).

Instead, [Dyer \(2009\)](#) used a [Maximum Entropy \(ME\)](#) model with universal features for [compound splitting](#) and generated [split](#) lattices (rather than using a single [split](#) decision) for improving [SMT](#). The background for using lattices is that there are several plausible valid splits (e.g., with respect to [splitting](#) depth or [word](#) sense ambiguity) and thus propagating the uncertainty about the [splitting](#) to downstream applications can be helpful (Dyer, 2009). [Figure 16.1](#) shows an example of a [split](#) lattice for the German [compound](#) *Ton|band|aufnahme* ‘tape recording’. For a translation to English, all paths except for [splitting](#) *aufnahme* are possible.

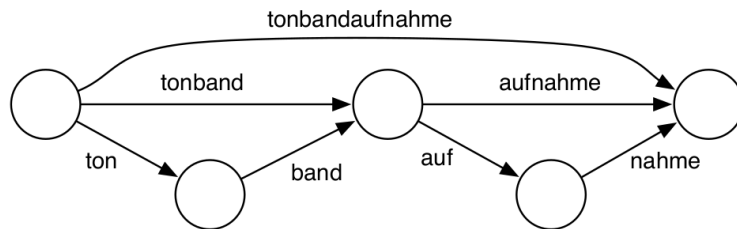


Figure 16.1.: Example of a split lattice in Dyer (2009)

The features used in the [ME](#) model include [constituent](#) frequency, [constituent](#) length, probabilities of [word](#)-initial  $N$ grams and the frequency of hypothesized [linking elements](#). Dyer (2009) showed that {German, Hungarian, Turkish}-to-English [SMT](#) systems can be improved by using [split](#) lattices, i.e., his [splitter](#) was [extrinsically](#) evaluated on the task of [SMT](#).

[Verhoeven et al. \(2014\)](#) used the hyphenation algorithm developed by Liang (1983) in a classification task. This algorithm has been implemented as a module in  $\text{\LaTeX}$ . It allows and disallows splits within certain letter combinations (Verhoeven et al., 2014). In the Dutch and Afrikaans dataset of Verhoeven et al. (2014), all [linking elements](#) are marked up. The [target languages](#) of (Verhoeven et al., 2014) were *Dutch* and *Afrikaans*. They performed an [intrinsic evaluation](#) on the predicted [constituent lemmas](#) (including [allomorphs](#)).



## 16.2. Linguistic Approaches

In contrast to statistical approaches (Section 16.1), linguistic approaches mainly rely on language-specific resources, such as a morphological analyzer or an extensive set of morphological transformation rules. While **compound splitters** including much linguistic information are designed for a specific **target language** and thus are less applicable to other **target languages**, they outperform statistical approaches in terms of **splitting** quality, as observed by Escartín (2014). In more recent work, Ziering and Van der Plas (2016) showed that statistical approaches are competitive to linguistic approaches concerning coverage and in addition the relative performance of statistical approaches is solid given that they are less dependent on language-specific resources.

### 16.2.1. Approaches based on a Morphological Analyzer

Instead of enumerating all possible analyses, morphologically informed methods are provided with an extensive morphological analysis (including inflectional, derivational and **compounding** information). These analyses are provided by morphological analyzers such as GERTWOL (Haapalainen and Majorin, 1995) or SMOR (Schmid et al., 2004). For **compounds** having several morphologically plausible **compound splits**, a ranking step based on additional information follows the initial analysis.

**Nießen and Ney (2000)** applied the morpho-syntactic analyzer GERTWOL on the German input of a German-to-English **SMT** system. Besides **compound splitting**, they also analyzed particle verbs (e.g., *los|fahren* ‘to set off’). For a morphological and syntactic disambiguation, they used the Constraint Grammar Parser for German, GERCG (Karlsson, 1990). For improving **compound splitting**, syntactic parsing can be helpful for disambiguating the **PoS** of the **compound head** (Cap, 2014, p. 137).

**Schiller (2005)** presented a **compound splitter** based on a **weighted Finite State Transducer** (wFST). The grounding is the exhaustive list of morphological analyses produced by the (non-weighted) finite-state Xerox morphological analyzer<sup>5</sup>. Weights are added such that **compound splits** with fewer, high-frequent **constituents** are promoted. The **splitter** is **intrinsically** evaluated based on the predicted **constituent lemmas**.

**Popović et al. (2006)** compared the approaches of Nießen and Ney (2000) with the statistical approach of Koehn and Knight (2003), outlined in Section 16.1, for **splitting** German **compounds** in German-to-English phrase-based **SMT**. Additionally, for the reverse direction (i.e., translating English text to German), prior **compound splitting** is

<sup>5</sup> <http://www.xrce.xerox.com/competencies/content-analysis/demos/german>

used for improving a [word alignment](#) model. Besides the problem of [undersplitting](#) due to a high [compound](#) frequency (as shown for *Aktionsplan* in Table 16.2), Popović et al. (2006) observed that a linguistic approach can produce splits for [compounds](#) whose [constituents](#) have no corpus evidence, for example the German [compound](#) *Arbeit|nehmer* ‘employee’. As a result, Popović et al. (2006) concluded that there is a similar improvement between linguistic and statistical [compound splitting](#) for German-to-English [SMT](#). Popović et al. (2006) focused on *German* as [target language](#).

**Fritzinger and Fraser (2010)** combined SMOR with the statistical approach of Koehn and Knight (2003) into a hybrid architecture. This method generates all morphologically plausible splits using SMOR and ranks them using the (log-based) geometric mean score of Koehn and Knight (2003). Any linguistic knowledge or restrictive parameters used in the initial version of Koehn and Knight (2003) are ignored within the hybrid approach, in which all necessary linguistic information is provided by SMOR. [Compounds](#) having three or more [constituents](#) are bracketed (or recombined) using training corpus lookup. Fritzinger and Fraser (2010) came to the result that the hybrid approach outperforms both individual methods in both [compound splitting](#) quality and [SMT](#) performance. For comparing the statistical approach of Koehn and Knight (2003) to the hybrid approach developed by Fritzinger and Fraser (2010), the following manually defined suffixes are added to the statistical method:  $\oplus$ nen,  $\oplus$ ien,  $\oplus$ en,  $\oplus$ er and  $\oplus$ n, as well as two truncatable suffixes:  $\ominus$ e and  $\ominus$ n. Fritzinger and Fraser (2010) focused on *German* as [target language](#). They performed an [intrinsic evaluation](#) on the predicted [constituent lemmas](#), as well as an [extrinsic evaluation](#) on the task of [SMT](#).

**Cap et al. (2014)** used [compound splitting](#) for training an English-to-German [SMT](#) system for translating English [open compounds](#) to German [closed compounds](#). Therefore, they adapted the approach of Fritzinger and Fraser (2010) for processing context-dependent tokens (e.g., for disambiguating named entities such as *Dinkelacker* (a beer brand vs. the [noun compound](#) ‘spelt field’)). Cap et al. (2014) focused on *German* as [target language](#). Her [splitter](#) was [extrinsically](#) evaluated using [SMT](#).

**Weller et al. (2014)** used SMOR as basis for investigating whether [splitting compounds](#) (or particle verbs) only in the case of compositionality is superior to an aggressive [splitting](#) strategy, when being applied prior to a German-to-English [SMT](#) system. They used compositionality scores derived from a [DSM](#) developed by Schulte im Walde et al. (2013). [Compounds](#) whose [constituents](#) (i.e., both [head](#) and [modifier](#)) have a low [Dsim](#) to the [compound](#) are considered as non-compositional. Weller et al. (2014) used two types of information for ranking several [split](#) options in the aggressive [splitting](#) base-

line: DIST (i.e., ranking splits according to the *Dsim*) and FREQ (i.e., ranking splits according to the corpus frequency, as proposed by Fritzinger and Fraser (2010)). They observed that there is no relevant difference between these information types for ranking splits, when evaluating on SMT. Weller et al. (2014) came to the conclusion that there is no improvement in phrase-based SMT performance when *splitting* only *compositional compounds*, because phrase-based SMT is robust to *oversplitting*: *oversplit compounds* (i.e., sequences of hypothesized *constituents*) can be learned as phrases. This is also in line with observations made by Dyer (2009) and Fritzinger and Fraser (2010). Weller et al. (2014) focused on *German* as *target language*. In their experiments, they used an *extrinsic evaluation* method based on SMT. While Weller et al. (2014) did not observe any difference between ranking *split* options according to frequency or *Dsim* (on the task of SMT), in Ziering et al. (2016), we observed that the *Dsim*-based ranking approach shows a clear improvement when being combined with frequency information and evaluated *intrinsically*, as will be discussed in Chapter 19.

### 16.2.2. Approaches based on Hand-crafted Transformation Rules

Adda-decker et al. (2000) manually defined a set of morphological rules for finding the correct *split point* (e.g., rules based on common German *modifier* suffixes, such as *-ungs*). Like Larson et al. (2000), Adda-decker et al. (2000) also evaluated their *compound splitter* *extrinsically* on the task of SR.

Although the approach of Weller and Heid (2012) is based on the statistical approach of Koehn and Knight (2003), we present it as a linguistic approach, because it contains a large amount of language-specific and hand-crafted information for German *compound splitting*. This decision is in line with Escartín (2014), who compared several linguistic and statistical *splitting* tools. Weller and Heid (2012) generated all possible splits and ranked them using the geometric mean score of the *constituent lemmas*' frequencies. They manually implemented an extensive list of morphological transformation rules for modeling *constituent inflection*. Moreover, *PoS* information serves for limiting *compound splitting* to *content words* and for restricting to splits where the *head*'s *PoS* equals the *compound*'s *PoS* (as suggested by Stymne (2008)). Weller and Heid (2012) focused on *German* as *target language*. As part of a bilingual term alignment task, Weller and Heid (2012) presented evaluation numbers of “good” splits on the first three ranks.

## 16.3. Performance of Splitting Approaches

### 16.3.1. Statistical vs. Linguistic Approaches

For summarizing the performance of all [compound splitting](#) approaches described in this chapter, we adopt the observations made by Escartín (2014), who compared the performance of various statistical and linguistic approaches on German [compound splitting](#).

Escartín (2014) qualitatively compared the statistical approach of **Popović et al. (2006)** against three linguistically motivated approaches: (1) the method developed by **Weller and Heid (2012)**, which relies on hand-crafted transformation rules, (2) the *BananaSplit* developed by **Ott (2006)** and (3) the [compound splitter \*jWordSplitter\*](#), developed by Daniel Naber<sup>6</sup>. A fourth linguistically motivated method, mentioned by Escartín (2014) as part of future work, is the SMOR-based system of **Fritzingler and Fraser (2010)**.

For a quantitative comparison, Escartín (2014) compared the statistical approach of **Popović et al. (2006)** against the linguistically motivated approaches of **Weller and Heid (2012)**. As [extrinsic evaluation](#), Escartín (2014) used the [SMT](#) system *Jane*, developed by Wuebker et al. (2012) for the task of German-to-Spanish [SMT](#). As development and test corpus, Escartín (2014) combined the TRIS corpus with EUROPARL. As evaluation measures, [BiLingual Evaluation Understudy \(BLEU\)](#), [Translation Edit Rate \(TER\)](#) and the number of cases of [Out-Of-Vocabulary \(OOV\)](#) were used. The results for both [splitters](#) and a baseline (no [splitting](#)) is given in Table 16.4.

Method	BLEU	TER	OOV
BASELINE	45.9%	43.9%	181
Popović et al. (2006)	48.3%	40.8%	104
Weller and Heid (2012)	48.3%	40.5%	114

Table 16.4.: SMT performance for different compound splitters

It turned out that “[splitting the compounds](#) improves the [BLEU](#) and [TER](#) scores and reduces the number of out of vocabulary words ([OOVs](#)) encountered” (Escartín, 2014). In general, the performance numbers for [SMT](#) were “very similar” (Escartín, 2014), leading to an [intrinsic evaluation](#) of the two [splitters](#).

As evaluation method, Escartín (2014) used Precision, Recall and Accuracy, as defined by Koehn and Knight (2003) and as will be described in Section 18.6.5. The results for

<sup>6</sup>[http://www.danielnaber.de/jwordsplitter/index\\_en.html](http://www.danielnaber.de/jwordsplitter/index_en.html)

Method	Precision	Recall	Accuracy
BASELINE	–	0%	89.79%
Popović et al. (2006) (TRIS+EUROPARL)	96.12%	72.51%	97.19%
Weller and Heid (2012)	99.23%	75.73%	97.49%

Table 16.5.: Intrinsic performance for different compound splitters

the two [splitters](#) are given in Table 16.5. For the approach of Popović et al. (2006), different training corpora were presented by Escartín (2014). For the sake of simplicity, we report only results for the best-working training corpus, i.e., TRIS + EUROPARL. It turned out that the linguistically motivated approach of Weller and Heid (2012) outperforms the statistical approach of Popović et al. (2006).

Escartín (2014) came to the conclusion that in general linguistically motivated approaches outperform statistical approaches, but this difference was not directly reflected in [SMT](#) performance but in an [intrinsic evaluation](#) method, as shown in Table 16.5. For using a statistical [splitting](#) approach on a technical corpus (e.g., the TRIS corpus), Escartín (2014) suggested to use both an in-domain corpus and large general data for getting best [splitting](#) performance.

### 16.3.2. Comparison in this Thesis

As will be described in Section 18.6.6, for the comparison of our [splitter](#) (which will be outlined in Chapter 18) against previous work on German [compound splitting](#), we decided to use linguistically motivated methods, because these provide the highest benchmarks.

Due to availability, we decided to use recent versions of the [compound splitters](#) of [Fritzinger and Fraser \(2010\)](#) (referred to as FF2010) and of [Weller and Heid \(2012\)](#) (referred to as WH2012). For comparing our method against previous work on Dutch and Afrikaans [compound splitting](#), we decided for the recent approach of Verhoeven et al. (2014). Since there is no implementation of their method available but their dataset, we will apply our method to their dataset and compare the results (for the accuracy of [split point](#) determination) with the performance numbers reported by Verhoeven et al. (2014) (referred to as VZDH2014). We do not compare to the most similar [multilingual](#) approach of Macherey et al. (2011), because that system is not available. Since Macherey et al. (2011) evaluated their method [extrinsically](#) on the task of [SMT](#), we cannot compare to published performance numbers (as will be done for

## 16. Related Work on Compound Splitting

VZDH2014). Performing an [extrinsic SMT](#)-based evaluation on our method (and compare the [BLEU](#) scores with the numbers published by Macherey et al. (2011)) fails, because Macherey et al. (2011) used [parallel](#) “non-public corpora”. We also tried to re-implement the [compound splitter](#) of Macherey et al. (2011), which has proved to be very time-consuming, even with the support of additional man-power. While the completion of this re-implementation exceeded the scope of this thesis, we plan to finish it in future work.

# 17. Morphological Operation Patterns

In this chapter, we present and elaborate parts of the work published in Ziering and Van der Plas (2016).

Macherey et al. (2011) described a representation of **compounding** morphology using a single character replacement at either the beginning, the middle or the end of a **word**. For our experiments, we adopt the format of Macherey et al. (2011) and elaborate it. Since it is possible that **constituent inflection** triggers morphological operations at several positions in a **word**, we combine all substring replacements into a pattern describing a series of operations. This transformation from a string  $\Sigma$  to a string  $\Omega$  is referred to as **Morphological Operation Pattern (MOP)**.

## 17.1. Compilation of MOPs

For compiling an **MOP**, we use the Levenshtein **Edit Distance (ED)** algorithm including the four operations INSERT (adding a character), DELETE (removing a character), REPLACE (exchanging a character  $\sigma_i$  by  $\omega_i$ ) and COPY (retaining a character). In a backtrace step, we determine the first possible sequence of operations that lead to a minimum **ED**. Except for COPY, we interpret all operations as replacements (INSERT and DELETE are replacements of or by an empty element  $\epsilon$  respectively). We merge all adjacent replacements by concatenating the source and target characters. Word-initial source and target sequences start with  $\wedge$  and word-final sequences end on  $\$$ . Sequences of adjacent COPY operations are represented by ‘:’ and separate the merged replacements. For example, in *Hühner|suppe*, the modifier lemma *Huhn* is transformed to *Hühner* by replacing u by ü (i.e., Umlautung) and adding the suffix er. The corresponding **MOP** is ‘u/ü:\$/er\$’. The second column in Table 17.1 shows some examples for the Germanic languages: *German*, *Dutch* and *Afrikaans*.

Language	MOP	Frequency	Examples
<i>German</i>	u/ü:\$/er\$	291	<Huhn, Hühner> ‘chicken’, <Buch, Bücher> ‘book’
	um\$/en\$	629	<Studium, Studien> ‘study’, <Medium, Medien> ‘medium’
<i>Dutch</i>	\$/en\$	7050	<arts, artsen> ‘doctor’, <band, banden> ‘tyre’
<i>Afrikaans</i>	\$/se\$	2768	<proses, prosesse> ‘process’

Table 17.1.: Examples of MOPs for *German*, *Dutch* and *Afrikaans*

## 17.2. Sources for MOPs

As described in Section 17.1, an **MOP** is compiled from a pair of strings  $\Sigma$  and  $\Omega$ . In this section, we present different ways of obtaining these string pairs, subject to **MOP** construction.

### 17.2.1. Word MOPs

**MOPs** can be learned automatically by exploiting the observation that **constituent inflection** and **word inflection** share a major part of the morphological operations. As a consequence, it is possible to approximate **constituent inflection** using **word inflection**, whose corresponding **MOPs** can be learned from a **lemmatized** corpus: for each **lemmatized** token (i.e., pair of **word form** and **lemma**), we determine the **MOP** that represents the transformation from **lemma** to **word form**. We collect all such **MOPs** with their corpus frequency. The third column in Table 17.1 shows the corpus frequencies for the **MOP** examples, acquired from general domain corpora in the respective languages. Subsequently, **MOPs** derived from **word inflection** are called **word MOPs**.

### 17.2.2. Gold-constituent MOPs

**MOPs** can be derived from a **compound splitting** gold standard (e.g., the GermaNet **compound** gold standard developed by Henrich and Hinrichs (2011)). For a **compound** of the form  $\alpha_{form}|\beta_{form}$  analyzed into the **lemmas**  $\alpha_{lem}$  and  $\beta_{lem}$ , the corresponding **MOPs** are derived from the string pairs  $(\alpha_{lem}, \alpha_{form})$  and  $(\beta_{lem}, \beta_{form})$ , i.e.,  $\text{MOP}[\alpha_{lem} \rightarrow \alpha_{form}]$  and  $\text{MOP}[\beta_{lem} \rightarrow \beta_{form}]$ . Subsequently, **MOPs** derived from gold **compound splits** are called **gold-constituent MOPs**.



### 17.2.3. Hand-crafted constituent MOPs

The transparent syntax of MOPs allows for a manual typing of the MOPs<sup>1</sup>. For some closed compounding languages, literature describes observed linking elements (such as an *er*-suffix), e.g., Langer (1998), who presented a German corpus study listing the 20 most frequent morphological operations observed for constituent inflection. While MOPs learned automatically using the Levenshtein ED algorithm already have the correct linear order, manually typed MOPs have to satisfy the linear order of substring replacements: WORD-INITIAL < WORD-INTERNAL < WORD-FINAL, whereas the WORD-INTERNAL replacements are ordered according to the position of the source string. Subsequently, MOPs which are manually implemented are called hand-crafted constituent MOPs.

## 17.3. MOP Application

MOPs can be used as a string manipulating function, i.e., as morphological transformation rules. Each substring replacement  $\mu_i$  can be applied to a string  $\Sigma$  sequentially ending in the string  $\Omega$ . However, some replacements  $\mu$  cannot be applied unambiguously on  $\Sigma$  and some replacements  $\mu$  cannot be applied at all (because there is no replacement source found in  $\Sigma$ ). Therefore, the MOP application is subject to a list of conventions:

- The application of a substring replacement  $\mu_i$  is done by replacing the source of  $\mu_i$  found uniquely in  $\Sigma$  by the target of  $\mu_i$ .

During MOP application,  $\Sigma$  is understood as starting with  $\wedge$  and ending on  $\$$ .

For example,  $\text{MOP}_{\$/er\$}(Kind) = Kinder$

- The substring replacements  $\mu$  are applied sequentially starting with the first.

For example,  $\text{MOP}_{u/ü:\$/er\$}(Huhn) = \text{MOP}_{\$/er\$}(Hühn) = Hühner$

- If the source of one replacement  $\mu_i$  is not part of  $\Sigma$ , the corresponding MOP cannot be applied and application returns NULL.

For example,  $\text{MOP}_{u/ü:\$/er\$}(Kind) = \text{NULL}$

- If the source of a replacement  $\mu_i$  is applicable at several positions, it is applied once for the last position. The motivation is that long words providing several options for a word-internal replacement  $\mu_i$  are compounds, where the last option has highest probability of being located in the head, which is subject to both

<sup>1</sup>As an alternative, hand-crafted constituent MOPs can be learned by manually providing constituent lemma-constituent form pairs.

word inflection and constituent inflection (i.e., the head of a complex compound modifier).

For example,  $\text{MOP}_{\underline{u}/\underline{ü}:\$/\text{er}\$}(S\underline{u}ppen\underline{h}u\underline{h}n) = S\underline{u}ppen\underline{h}\underline{ü}h\underline{n}er$  ‘boiling hen’ or

$\text{MOP}_{\underline{a}/\underline{ä}:\$/\text{e}\$}(P\underline{a}tent\underline{a}n\underline{w}\underline{a}l\underline{t}) = P\underline{a}tent\underline{a}n\underline{w}\underline{ä}l\underline{t}e$  ‘patent attorney’

This way, MOP application should work correctly for most cases and can be used for the normalization of constituent forms during compound splitting.

## 17.4. Inverting MOPs

By default, the direction of the MOP compilation introduced in Section 17.1 is from lemma to the word form, i.e.,  $\text{MOP}[Huhn \rightarrow Hühner] = \underline{u}/\underline{ü}:\$/\text{er}\$$ . However, for the normalization of a constituent form using MOP application during compound splitting, the direction is reversed, from word form to lemma. For this task, the MOPs acquired from any of the sources described in Section 17.2 are inverted by swapping the source and target of each substring replacement  $\mu_i$ .

For example, the MOP transforming *Huhn* to *Hühner* is  $\text{MOP}[Huhn \rightarrow Hühner] = \underline{u}/\underline{ü}:\$/\text{er}\$$ . Inverting this MOP results in:  $\text{MOP}[Hühner \rightarrow Huhn] = \underline{ü}/\underline{u}:\text{er}\$/\$$ . Re-applying the inverted MOP to *Hühner* results in the initial lemma:  $\text{MOP}_{\underline{ü}/\underline{u}:\text{er}\$/\$}(Hühner) = Huhn$ .

# 18. Multilingual Compound Splitting

In this chapter, we present and elaborate parts of the work published in Ziering and Van der Plas (2016).

The **multilingual compound splitter** we propose in this thesis can be considered as a statistical approach (see Section 16.1). It exploits knowledge about **word inflection**, encoded as **Morphological Operation Patterns (MOPs)** (see Chapter 17). Our **splitting** method can process **compounds** composed of any number of **constituents**<sup>1</sup>. The **constituents' word** category have to be any **content word** type. **Lemmas** which are **function words** are excluded as **constituents**<sup>2</sup>. Subsequently, this restriction will be referred to as **constituent content word restriction**. The **splitter** provides both the **split point** information (e.g., *Hühner* | *suppe*) and the **normalized constituents** (e.g., *Huhn* + *Suppe*). It is designed recursively, which allows for representing the **compound split** both hierarchical (i.e., as a **split tree** structure) and as a linear sequence.

Figure 18.1 shows the architecture of the **splitting** algorithm. The recursive main method starts with the **target compound** as a single **constituent** and recursively splits the **constituents** produced by the **binary splitter** (Section 18.1) until an **atomic** analysis is returned. The **splitting** algorithm performs a **recursive lemma splitting**, i.e., for the recursive application of the **binary splitter**, the **normalized constituents** are used (e.g., after **splitting** and **normalizing** *Suppenhühner* | *zucht* ‘boiling hen breeding’ into the **constituent lemmas** *Suppenhuhn* and *Zucht*, the **binary splitter** is applied to *Suppenhuhn* instead of on *Suppenhühner*). The **binary splitter** generates all possible **split points** for the **target**. In the next step, the **constituent forms** resulting from all **split points** are **normalized** (Section 18.2), all candidate **lemma** combinations are ranked and the best **split** is returned (Section 18.3).

---

<sup>1</sup>The hierarchical and **binary splitting** architecture requires that the **immediate constituents** at each level must be able to be mapped on known corpus **lexemes**.

<sup>2</sup>This prevents oversplits into functional **head lemmas**, e.g., *Bohrer* ‘driller’ would be **split** into *bohr* | *er* ‘drilling he’.

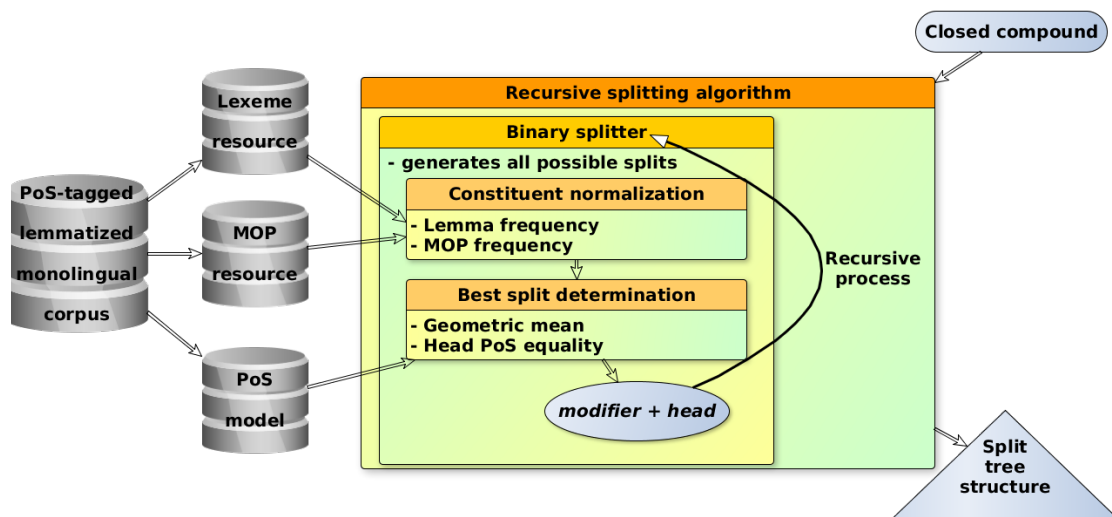


Figure 18.1.: Architecture of the splitting algorithm

## 18.1. Binary Splitter

First, all possible **binary** splits are generated (i.e., all possible **split points**) with a minimum **constituent form** length of 2 characters (e.g., for *Ölpreis* ‘oil price’, the system generates *Ö|lpreis*, ..., *Ölpre|is*) and a **non-split option** (i.e., an **atomic** analysis without **split point**) is added. In the next step, all **constituent forms** resulting from these analyses are **normalized**, i.e., the underlying corpus **lemmas** are retrieved using a list of previously collected **lemmas** including information such as corpus frequency or **PoS** tags. Subsequently, we will call this resource the **Lemma Resource (LR)**. The system retrieves the  $M$  most probable **lemmas** based on a **lemma score**, as described in Section 18.2. In the final step, all  $M^2 + 1$  **lemma** combinations derived from all **binary** splits (and the **non-split option**) are ranked according to a **combination score** described in Section 18.3. The highest-ranked **split** is returned.

### 18.1.1. Split Point Markers

A special case of **compounds** are those with a **split point marker**. There are different types of **split point markers**. Usually, it is a special character (which is not part of the **constituent alphabet**) marking the boundary between two **constituents**, i.e., the **split point**. The (almost exclusive) representative of a **split point marker** is the hyphen as in

*TV-Programm* ‘TV program’. Alternative [split point markers](#)<sup>3</sup> are the pipe symbol (as commonly used in the [split point format](#) (SPF), e.g., *Hühner|suppe* ‘chicken | soup’), a slash symbol (indicating a conjunction, e.g., *Wohn/Ess-zimmer* ‘living-dining room’ or *CDU/CSU-Fraktion* ‘CDU / CSU group’) or implicit as a case transition or **camel case** (i.e., the transition from lowercase to uppercase, often used in names as the German *BahnCard* ‘rail card’). When the [binary splitter](#) identifies a [split point](#) marked **compound**, it only generates [binary](#) splits at these markers, i.e., for *Science-Fiction-Film* ‘science fiction movie’, two [binary](#) splits are generated: *Science* and *Fiction-Film*, and *Science-Fiction* and *Film* (i.e., [splitting](#) is broken down to [bracketing](#)). Moreover, the [non-split option](#) is not added, because one of the possible [binary](#) splits are considered to be correct. The [constituent form](#) length limitation is not relevant for [split point](#) marked **compounds**; these **compounds** are always [split](#) (e.g., *C-Dur* ‘C major’). This way, the [binary splitter](#) is able to also decompose [split point](#) marked **compounds** into possibly unknown [constituents](#). If all [binary](#) splits based on [split point markers](#) have a zero score, the split with the right-most [split point marker](#) is selected by default. As will be shown in Section 24.4,  $N$ -partite ( $N \geq 3$ ) **compounds** are mostly structured in a LEFT-branched tree (see [LEFT class baseline](#)).

## 18.2. Constituent Normalization

This subtask addresses the [normalization](#) of a [constituent form](#). As part of the [binary splitter](#), the strings subject to [normalization](#) are not necessarily well-formed [constituents](#) (e.g., *Ölpr* derived from the false [split](#) *Ölpr|eis*). The system tries to find the most probable [lemma](#) given the assumption that the provided string is a valid [constituent form](#). Having only low confidence scores for all retrieved [lemmas](#) indicates a low probability of having the correct [constituent form](#).

### 18.2.1. Ngram Index Lookup

In Ziering and Van der Plas (2016), we used an  $N$ gram index for [constituent normalization](#). In this [Lemma Resource](#) (LR),  $N$ grams of corpus [lemmas](#) are mapped onto a list of frequency distributions of corresponding corpus [lemmas](#), sorted by [lemma](#) length. The goal was to collect the most frequent corpus [lemmas](#) sharing most  $N$ grams with a

<sup>3</sup>The current set of [split point markers](#) is language-independent and contains: -, +, |, #, \_, ., :, /, &. But in general, all characters not included in the [constituent alphabet](#) can function as [split point marker](#).

given [constituent form](#). This way, [lemmas](#) could be looked up efficiently, while allowing any possible morphological transformation. In a final step, [word MOPs](#), as described in Section 17.2.1, are used for filtering noisy [lemmas](#). While this approach shows solid performance in [constituent normalization](#), in this thesis, we decided to use a much more scaling alternative for [normalization](#) which includes [MOP application](#) (17.3) of [word MOPs](#), on [constituent forms](#) directly. We will explain this in Section 18.2.2.

### 18.2.2. MOP Application

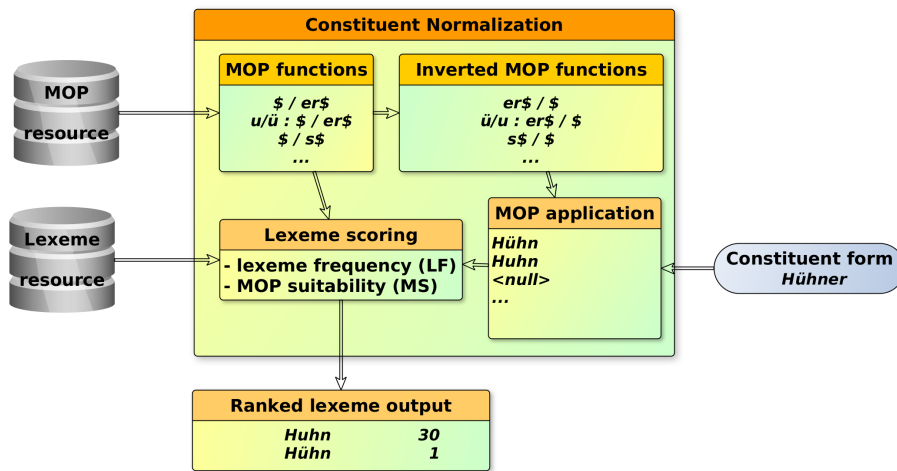


Figure 18.2.: [Constituent normalization](#) by using [MOP application](#)

Figure 18.2 shows the architecture of the [constituent normalization](#) using [MOP application](#). The [MOP Resource \(MR\)](#) contains a list of [MOPs](#) which are inverted prior to [MOP application](#), as described in Section 17.4. All inverted [MOPs](#) are applied to the [constituent form](#)  $c$ , as described in Section 17.3, yielding several candidate [lemmas](#) (possibly NULL). The candidate [lemmas](#) are scored using the type-based frequency of the respective original [MOP](#) (derived from the [MR](#)) and using the frequency distribution of corpus [lemmas](#) (derived from the [LR](#)). The  $M$  top-ranked [lemmas](#) are returned.

In analogy to Figure 18.2, the pseudocode for the [MOP application](#)-based [normalization](#) is given in Algorithm 18.1. As input, the algorithm takes as [LR](#) a list of [lemmas](#) and associated corpus frequencies and as [MR](#) a list of [MOPs](#) with associated frequency. All retrieved candidate [lemmas](#) are stored together with its score in the list [CLs](#) (line 1). For a given [constituent form](#)  $c$ , the system inspects all collected [MOPs](#)  $MOP_i$ . Firstly,  $MOP_i$  is inverted (i.e., the source and target substrings of each replacement  $\mu$

---

**Algorithm 18.1** MOP application-based lemma lookup

---

**Target:** Constituent form  $c$ **Input 1:** Lemma Resource (LR): List of lemmas and associated corpus frequency**Input 2:** MOP Resource (MR)

```

1: CLs  $\leftarrow \{ \}$  {the candidate lemma (mapping lemmas to the score)}
2: for  $MOP_i$  in MR do
3:    $IM_i \leftarrow invert(MOP_i)$ 
4:    $l_i \leftarrow IM_i(c)$ 
5:   if  $l_i \neq \text{NULL}$  then
6:      $CLs[l_i] = score(l_i, LR, MOP_i, MR)$ 
7:   end if
8: end for
9: rank(CLs)
10: return top $M$ (CLs)

```

---

is swapped) into  $IM_i$  (line 3). By default,  $MOP_i$  represents the transformation from lemma to constituent form. The inverted MOP changes the direction, from constituent form to constituent lemma.  $IM_i$  is applied to  $c$  (line 4). If the application result is not NULL (cf. Section 17.3), the potential lemma  $l_i$ , which gets a score according to a lemma scoring method outlined below, is stored together with its score in CLs (lines 5-6). All candidate lemmas are ranked and the top  $M$  candidates are returned (lines 9-10).

**Lemma Scoring** The lemma scoring for a lemma  $l_i$  is based on two features: (1) the lemma corpus frequency  $LF(l_i) = cf(l_i)$  and (2) the MOP Suitability (MS), as given in Formula 18.1.

$$MS(l_i) = \frac{\log_{10}(freq(\text{MOP}[l_i \rightarrow c]))}{ED[l_i \rightarrow c] + 1} \quad (18.1)$$

The MS estimates the suitability of using the MOP transforming  $l_i$  for the constituent form at hand,  $c$ , (represented as  $\text{MOP}[l_i \rightarrow c]$ ) as a candidate for constituent inflection. As the first component, we use the  $\log_{10}$  of the frequency associated with the MOP,  $\log_{10}(freq(\text{MOP}[l_i \rightarrow c]))$ . This value is rescaled with the resulting ED between the candidate lemma  $l_i$  and the constituent form,  $c$ , (represented as  $\text{ED}[l_i \rightarrow c]$ ). For avoiding a division by zero, we perform add-1 smoothing. The rescaling is motivated by the fact that MOPs having a small ED are more prominent in compounding. Such MOPs are not necessarily most frequent in word inflection. For example, one of the most frequent word MOPs is derived from the irregular Afrikaans verb inflection of *wees* ‘to be’, having the verb form *is* and thus leading to the high-frequent word MOP  $\text{MOP}[wees \rightarrow is] = \hat{w}ee/\hat{i}$

with a large ED of 3. Therefore, the ED rescaling demotes the suitability score of such word MOPs for constituent inflection.

The final lemma score is a product of LF and MS, as given in (Formula 18.2).

$$score(l_i) = LF(l_i) \cdot MS(l_i) \quad (18.2)$$

### 18.3. Best Split Determination

In the final step, the best split option among all lemma combinations (i.e., pairs of retrieved candidate lemmas for modifier ( $l_m$ ) and head ( $l_h$ ), and corresponding split point) and the non-split option is determined. For this task, a combination model, which considers the interaction<sup>4</sup> between  $l_m$  and  $l_h$  is used.

Inspired by Koehn and Knight (2003), as a first feature, the geometric mean of the lemma scores, as given in (Formula 18.3), is used. For binary splits, the constituent set  $con = \{l_m, l_h\}$  is used and for the non-split option,  $con = \{l_h\}$ .

$$geo(con) = \sqrt[|con|]{\prod_{l_i \in con} score(l_i)} \quad (18.3)$$

The second feature of the combination model is motivated by the RightHand Head Rule (RHHR), saying that the compound head is the right-most constituent and encodes the principal semantics and the PoS of the whole compound (at least for endocentric compounds, as discussed in Section 3.7). As done by previous splitting approaches (Stymne, 2008, Weller and Heid, 2012), we assume the RHHR and allow only splits for which the righthand side constituents have the same PoS as the compound. Since the compound splitter presented in this thesis works out of context, we try to subsume all possible readings (also meaning all possible PoS tags) by representing them as a distribution over the PoS probabilities  $p(\text{PoS}|\text{word}) = \frac{freq(\text{PoS} \cap \text{word})}{freq(\text{word})}$  acquired from the monolingual PoS-tagged corpus. The value of the head-PoS-EQuality (hEQ) feature is defined as the cosine similarity between the PoS probability distributions of compound  $\Psi$  and compound head  $l_h$ ,  $hEQ(\Psi, l_h)$ . If the PoS tag of the compound is unknown, we use a default cosine value of 1.0.

$$split(con) = geo(con) \cdot hEQ(\Psi, l_h) \quad (18.4)$$

<sup>4</sup>For example, a split option with a highly scored head can be demoted by a very low score for the modifier, or vice versa.



Finally, all candidate **lemma** combinations (including the **non-split option**) are ranked according to the **splitting** score given in Formula 18.4. The highest-scored **split** is returned as output of the **binary splitter**, being subject to the recursive process. Figure 18.3 shows an example of the recursive **splitter** output for the German **compound** *Studienbescheinigungsablaufdatum* ‘enrollment certification expiration date’ with the related **MOPs**.

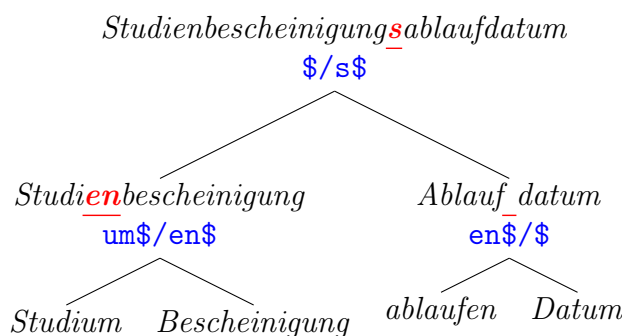


Figure 18.3.: Example of a **split tree** structure with related **MOPs**

## 18.4. Additional Compound Splitting Features

In the presented approach, **word inflection** is used as an approximation for **constituent inflection** by compiling **word MOPs** from **lemmatized** corpora and inversely apply these to potential **constituent forms**. However, this approximation leads to operations which are exclusively valid for **word inflection**. In this section, we present some restrictive features<sup>5</sup> for mitigating the impact of noisy **word MOPs** on **compound splitting**. In order to keep the presented **compound splitting** approach as independent from language-specific information as possible, these features are designed universally for being flexibly applicable to a large variety of **compound splitting** languages.

### 18.4.1. Prior MOP Lemmatization

When applying a **compound splitter** to a running text (instead of on a list of **lemmatized** units), there are often inflected **word forms** to be **split** (e.g., the pluralized German noun *Termine* ‘appointments’). Although the **compound splitter** is able to map an inflected **word form** on a **lemma** (i.e., by using the **non-split option** and **normalizing** the single

<sup>5</sup>In Ziering and Van der Plas (2016), we did not use these features.

constituent), there is a high risk of using a misleading **word MOP** that yields a false **binary split**. For example, *Termin* is **split** into *Ter | mine* and **normalized** to *Tor Mine* ‘goal mine’ using the **MOP** *o/e* which is valid for nouns like *Stadion* ‘stadium’ having the plural form *Stadien* ‘stadiums’. In contrast, the **lemma** *Termin* ‘appointment’ is correctly analyzed as an **atomic** unit.

Therefore, a **target** subject to **compound splitting** undergoes a **lemmatization** step before. For this purpose, we exploit the potential of **MOPs** as a rule-based and context-free **lemmatizer**. We add all observed **word MOPs** to the respective **lemma** in the **LR**. For example, the **lemma** *Termin* lists the **word MOPs** shown in Table 18.1.

<b>Word MOP</b>	<b>Word form</b>	<b>Number</b>	<b>Case</b>		
<b>_/_ (null-MOP)</b>	<i>Termin</i>	singular	nominative	accusative	dative
<b>\$/s\$</b>	<i>Termins</i>		genitive		
<b>\$/es\$</b>	<i>Termines</i>				
<b>\$/e\$</b>	<i>Termin</i>	plural	nominative	accusative	genitive
<b>\$/en\$</b>	<i>Terminen</i>		dative		

Table 18.1.: Word MOPs for the **lemma** *Termin* ‘appointment’

The **lemmatization** of a **word** corresponds to the **normalization** (described in Section 18.2) of the **word** (i.e., as a **constituent form**) using the restriction of a **lexeme agreement**, as described below.

**Lexeme agreement:** For a **word MOP** ( $\text{MOP}_\alpha$ ), the inversion ( $\text{MOP}_\alpha^{\text{INV}}$ ) may only be applied to a **constituent form** if the original **word MOP** ( $\text{MOP}_\alpha$ ) is listed in the resulting **lexeme’s** entry of the **LR**.

If the **word** subject to **split** cannot be mapped on a listed corpus **lemma** using **lexeme agreement** restricted **normalization**, the **compound splitter** is applied to the original **word**, otherwise the retrieved corpus **lemma** is used. For example, *Termin* is **normalized** to *Termin* by applying the inversed **MOP** *e\$/*, because the original **MOP**(*\$/e\$*) is listed in the **LR**-entry of *Termin*. For the final representation of the **compound split** (e.g., in the **split point format** (**SPF**)), the original **word inflection** is re-added.

### 18.4.2. Compound Content Word Restriction

Although most **function words** are short and have a lower risk of being **split**, there are some cases that can be considered as a composition, but should actually not be **split**,

because they are used as an opaque unit (e.g.,, *ob|wohl* ‘although’, *so|bald* ‘as soon as’, *da|nach* ‘afterwards’, *da|von* ‘thereof’, ...).

For avoiding the **splitting** of such **function words**, the **binary splitter** rejects functional **targets** and always returns an **atomic** analysis. After **lemmatizing** the **target** (as described in Section 18.4.1), the **PoS** (as listed in the **PoS** probability distribution of each **LR**-entry) is determined. If the majority (> 50%) of all observed **PoS** tags points to a **content word**, the **target** is further analyzed by the **binary splitter**, otherwise it is kept unsplit.

### 18.4.3. PoS Agreement Restriction for the Modifier

In some cases, a non-compound gets falsely **split**, because the **target**’s prefix can be **normalized** to a **lemma** whose **PoS** is different from that **PoS** which is related to the **lemma** used for learning the applied **MOP**. For example, the non-compound *Quartal* ‘quarter (year)’ is falsely **split** as shown in Figure 18.4, i.e., into the adverb *quer* ‘across’ and the noun *Tal* ‘valley’. The causing **MOP**, **e/a**, is learned from verbs like *sprechen* ‘to speak’, having the past tense form *sprachen* ‘spoke’.

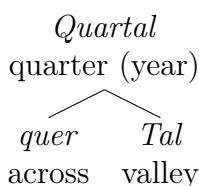


Figure 18.4.: False compound split of *Quartal* ‘quarter (year)’

Using **MOP application** with a **PoS** agreement (similar to the **lexeme** agreement described in Section 18.4.1), it is possible to reduce erroneous splittings which are due to cross-categorical **MOP application**.

Therefore, for all **MOPs** in the **MR**, the observed **PoS** of the related **word forms** are collected. Moreover, all observed **PoS** for each **lexeme** are stored in the **LR**. **MOP application** is restricted by a **PoS agreement**, as described below.

**PoS agreement:** For a **word MOP** ( $\text{MOP}_\alpha$ ), the inversion ( $\text{MOP}_\alpha^{\text{INV}}$ ) may only be applied to a **constituent form** if there is at least one common **PoS** between the collection for the original **word MOP** ( $\text{MOP}_\alpha$ ) and in the **PoS** set of the resulting **lexeme**.

Since the **MOP** **e/a** has no evidence for adverbs, it cannot be applied to *quer* and the **word** *Quartal* is kept unsplit.

#### 18.4.4. Lexeme Agreement Restriction for the Head

Most closed compounding languages realize constituent inflection not on both constituent types, but only on one of them, for example, Germanic languages have constituent inflection only on the modifier. Although the knowledge about which constituent type undergoes constituent inflection is language-specific and should therefore be excluded from an absolutely language-independent splitting method, we decided to include it in the proposed MOP-based compound splitter, because this information, which is included in most language-specific previous splitting methods, is very minor compared to extensive knowledge about morphological operations. In fact, just one out of three classes is necessary:

1. **modifier inflection** - constituent inflection is applied only to modifiers
2. **head inflection** - constituent inflection is applied only to the head
3. **both inflection** - constituent inflection is applied to both modifiers and head

For languages in which only the modifier undergoes constituent inflection (e.g., Germanic languages), the head still undergoes word inflection (i.e., pluralization, case-marking, ...). In order to split word-inflected (e.g., pluralized) compounds correctly, the MOP-based lemmatization (described in Section 18.4.1) can be applied to the head.

For example, the lemma *Lauf* ‘run’, as in *Marathonlauf* ‘marathon run’, lists the word MOPs shown in Table 18.2.

Word MOP	Word form	Number	Case		
_/ _ (null-MOP)	<i>Lauf</i>	singular	nominative	accusative	dative
\$/s\$	<i>Laufs</i>		genitive		
\$/es\$	<i>Laufes</i>		genitive		
\$/e\$	<i>Laufe</i>		dative		
a/ä:\$/e\$	<i>Läufe</i>	plural	nominative	accusative	genitive
a/ä:\$/en\$	<i>Läufen</i>		dative		

Table 18.2.: Word MOPs for the lemma *Lauf* ‘run’

The frequent word MOP a/ä:\$/er\$ (as in <*Mann,Männer*> ‘<man,men>’) does not occur in the paradigm of *Lauf*. Splitting the compound *Marathonläufer* ‘marathon runner’ using all collected word MOPs yields the lemmas *Marathon* and *Lauf* due to the higher frequency of *Lauf*, whereas the restriction to a lexeme agreement for the head

yields the correct lemmas *Marathon* and *Läufer* derived by the null-MOP ( $\_/\_$ ), which is valid in the paradigm of *Läufer*.

### 18.4.5. Trade-off between PoS and Lexeme Agreement

The presented way of using a PoS agreement on the modifier and a lexeme agreement on the head is not optimal. The PoS agreement restriction for the modifier is too lenient. As described in Section 18.4.4, *Läufer* ‘runner’ can be falsely normalized to *Lauf* ‘run’ due to the word MOP  $a/\ddot{a}:\$/er\$  (which is valid for nouns like *Mann* ‘man’). Since both *Lauf* and *Mann* are nouns, the PoS agreement cannot prevent a false normalization of the modifier in compounds like *Läuferteam* ‘runner’s team’ as shown in Figure 18.5.

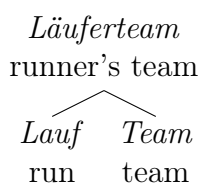


Figure 18.5.: False compound split of *Läuferteam* ‘runner’s team’

However, applying the lexeme agreement restriction to the modifier would be too restrictive, because there are many lexemes that do not share MOPs from word inflection and constituent inflection, e.g., nouns ending on *-heit* (e.g., *Kindheit* ‘childhood’, *Schönheit* ‘beauty’ or *Menschheit* ‘manhood’): while *Kindheit* gets *s*-suffixed as a modifier (as in *Kindheits|erinnerung* ‘childhood memory’), *Kindheits* is not a paradigmatic word form and thus the word MOP  $\$/s\$  is not listed in the LR-entry of *Kindheit*. Therefore, the MOP application has to generalize over the lexeme.

Finding an intermediately restricted level between PoS agreement and lexeme agreement will be addressed in future work.

## 18.5. Compound Splitting Representation

The compound splitter presented in this chapter recursively splits constituents of a compound as long as a binary analysis has a higher score than the non-split option i.e., the analysis as an atomic constituent.

### 18.5.1. Split Tree

The recursive **splitting** architecture allows for a hierarchical representation of the **binary compound splits**. Therefore, the system produces a **split tree** containing all **binary splits**. Each branch has a score corresponding to the **split score** (outlined in Formula 18.4) of the respective **binary split**.

#### Flexible Tree Pruning

The granularity of the morphological analysis (or **splitting depth**) which is necessary differs with the type of application. For **MT**, a **compound** does not need to be **split** deeper than into **constituents** for which a translation is known, whereas for linguistic research, a deeper morphological analysis is desirable. For example, an exhaustive **split tree** of the **compound** *Schauspielzeitschrift* ‘drama journal’ is given in Figure 18.6.

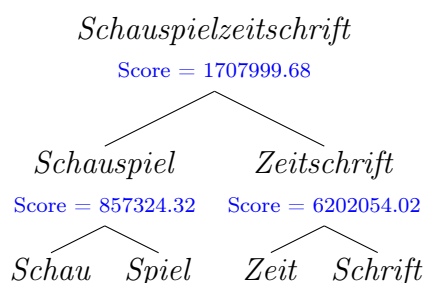


Figure 18.6.: Linguistically motivated split of *Schauspielzeitschrift*

While the **constituents** *Schauspiel* ‘drama’ and *Zeitschrift* ‘journal’ do not need to be **split** for the task of **MT**, for a linguistic analysis a **split** into all four **atomic constituents** is also valid and introduces etymological clues (e.g., in how far is *Zeit* ‘time’ related to *Zeitschrift* ‘journal’).

Neither the **compound splitter** presented in this thesis nor the gold standards for **intrinsic evaluation** (Section 18.6.4) are designed for a specific task in mind. For example, the **compound splitting** gold standard developed by Henrich and Hinrichs (2011) only provides **binary splits**, even for **compounds** with three or more plausible **atomic constituents** (e.g., *Fußball|länderspiel* ‘international football game’).

For catering for all **NLP** tasks, the output of the presented **compound splitter** provides a **split tree** for all numbers of **leaf nodes**  $1 \leq k_i \leq k_{init}$ , where  $k_{init}$  is the number of **leaf nodes** for the initial **split tree** produced by the recursive **compound splitting** method. Therefore, a new **split tree** is iteratively created where one branch into two **leaf nodes** is removed. If there are several leaf branches, the one with the least score is pruned. For

example, the **four-noun Compound (4NC)** given in Figure 18.6 is pruned to the trees presented in Figure 18.7.

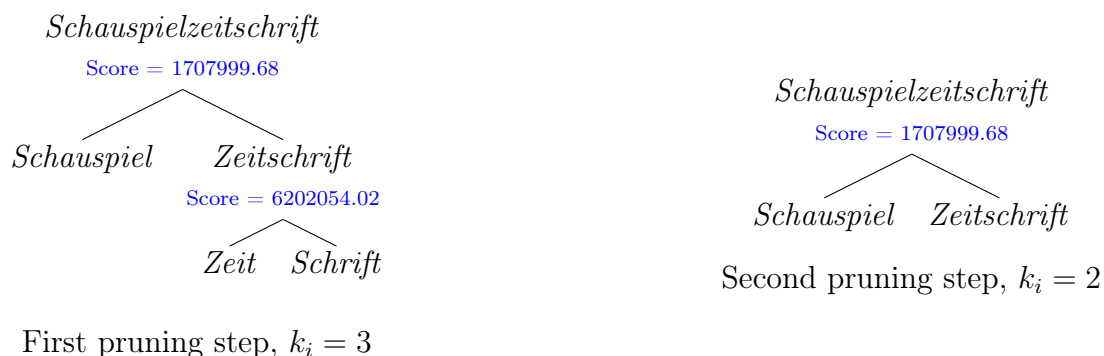


Figure 18.7.: Tree pruning for *Schauspielzeitschrift*

In order to meet the standards of an underlying gold standard,  $k_i$  is set to the numbers of gold **constituents**,  $k_{gold}$  (e.g., to 2 for the binary splits in the gold standard from Henrich and Hinrichs (2011)).

We are aware of the fact that it is unrealistic to determine the number of **constituents** prior to **compound splitting** and that **oversplitting** cannot be measured this way. For measuring the degree of **oversplitting** and **undersplitting**, we present an **extrinsic evaluation** method in Chapter 20, where the deepest **splitting** analysis (i.e.,  $k_i = k_{init}$ ) is used.

Since no **compound splitting** gold standard provides a hierarchical representation of the **compound splitting**, but a linear sequence of **constituents**, the **split tree** is flattened into a linear representation (i.e., the information of all **leaf nodes** are collected and sequentially concatenated).

For evaluating **splitting** performance in the disciplines of (1) **split point** determination and (2) **constituent normalization** (as discussed in Section 15.1.2), the system creates two linear **splitting** representations: the **lemma sequence format (LSF)** and the **split point format (SPF)**.

### 18.5.2. Linear Lemma Sequence Format

The **lemma sequence format (LSF)** lists all **constituent lemmas** separated by space. For example, the **splitting** of the **compound** *Hühnersuppenrezept* ‘chicken soup recipe’ is represented in the **LSF** as: *Huhn Suppe Rezept*.

For catering **NLP** tasks in which information about morpho-syntactic features such as case or number are relevant (e.g., for the subject-verb agreement in syntactic sentence

parsing), we use an additional LSF format which contains **constituent lemmas** only with respect to **constituent inflection** and keeps the **word-inflected head**. Subsequently, this format will be referred to as  $\text{LSF}_{\text{word}}$ . For example, the  $\text{LSF}_{\text{word}}$  for the pluralized *Hühnersuppenrezepte* ‘chicken soup recipes’ is *Huhn Suppe Rezepte*. A further motivation for using the  $\text{LSF}_{\text{word}}$  is the fact that there are gold standards or **compound splitting** systems that provide **normalized constituents** except for the **word inflection** at the **head**. For evaluating on such gold standards and for comparing against such **compound splitting** systems, the  $\text{LSF}_{\text{word}}$  is used.

### 18.5.3. Linear Split Point Format

For measuring the performance of determining the correct **split point**, the **split point format (SPF)** lists all **constituent forms** (without any true-casing) separated by the pipe symbol. For example, the **splitting** of the **compound** *Suppenhühnerzucht* ‘boiling hen breeding’ is represented in the **SPF** as: *Suppen | hühner | zucht*.

For some reasons (e.g., the recursive **lemma splitting**, described in the beginning of Chapter 18, or the fact that some **compound splitters** only provide **LSFs**), the **SPF** has to be compiled. This compilation is a non-trivial task. We developed several methods for **SPF** compilation and conducted experiments with them. Discussing all methods in detail would go beyond the scope of this chapter. Nonetheless, we present the methods and the different experiments in Appendix C.

## 18.6. Experiments

In this section, we conduct several experiments that will measure the performance of the proposed **compound splitting** method and help to answer some of the research questions posed in Section 15.2. The following description of the evaluation setup subsumes parts of all experiments conducted within the following evaluation blocks.

### 18.6.1. Target Languages

In the following experiments, we evaluate **splitting** performance on three **closed compounding languages**: *German* (Section 3.9.2), *Dutch* (Section 3.9.3) and *Afrikaans* (Section 3.9.4). While the **compound splitter** is supposed to show comparable performance in many more **target languages** (in particular in those in which **constituent inflection** and **word inflection** share a substantial amount of morphological operations), there are



gold standards for [compound splitting](#) available for only a few languages. An [intrinsic evaluation](#) is only possible for those languages that provide a [compound splitting](#) gold standard.

## 18.6.2. Training Data

### Corpora

For the three [target languages](#) listed in Section 18.6.1, we used the corpora derived from WIKIPEDIA<sup>6</sup>.

It is well-known in [NLP](#) that the bigger the training data and the better the quality of preprocessing (i.e., of the meta data such as [lemmas](#) or [PoS](#) tags), the better the performance of the respective [NLP](#) task. This is also true for [compound splitting](#). Large and well preprocessed corpora provide corpus [lemmas](#) with representative frequencies, which are used for looking up the [constituent lemmas](#). Large corpora also provide information for compiling a representative set of [MOPs](#) describing the [word inflection](#) in the respective [target language](#). Escartín (2014) compared different types of [compound splitters](#) and showed that corpus-based [compound splitting](#) methods perform much better on large corpora.

On the other hand, all [compound splitting](#) systems and setups discussed in this thesis are trained on the same Wikipedia corpus (with respect to a certain [target language](#)). Thus, the discussed comparisons are meaningful and help to point out the advantages and disadvantages of the different systems.

### Preprocessing

For [tokenizing](#), [PoStagging](#) and [lemmatizing](#) the German and Dutch corpora, the Tree-Tagger<sup>7</sup> (Schmid, 1995) has been used. It is important to use a [lemmatizer](#) model that conforms with the latest orthography rules, which are supposed to be the basis of all [compound splitting](#) gold standards used in the experiments presented below. For example, according to the German orthography reform<sup>8</sup> of 1996, a short stressed vowel is never followed by the grapheme ‘ß’ (e.g., *Kuß* ‘kiss’ (old spelling) vs. *Kuss* (new spelling according to the reform)). A mismatch between the gold standard’s orthography and the orthography used for preprocessing the training corpora can lead to noisy

---

<sup>6</sup>[{de,nl,af}.wikipedia.org](#)

<sup>7</sup>[www.cis.uni-muenchen.de/schmid/tools/TreeTagger](#)

<sup>8</sup>[en.wikipedia.org/wiki/German\\_orthography\\_reform\\_of\\_1996](#)

and incorrect German **word MOPs**. For example, while the **MOP** for pluralizing *Kuss* (cf. *Küsse* ‘kisses’) is very general and valid for many other **lexemes** (i.e., **u/ü:\$/e\$**), the **MOP** for pluralizing *Kuß* would be very specific and less applicable to other cases (i.e., **uß\$/üsse\$**). Moreover, a German **Lemma Resource (LR)** that is out of date, cannot be used reliably for string matching with **constituent lemmas** occurring in the gold standards (e.g., *kussecht* ‘kissproof’ → *Kuss* + *echt*).

The **tokenization** of the Afrikaans corpora has been performed with the method described in Augustinus and Dirix (2013). Afrikaans **PoS** tagging has been done using the tool described in Eiselen and Puttkammer (2014) and for **lemmatization**, the system of Peter Dirix has been utilized, the second author of the previous paper.

Table 18.3 shows some statistics of the selected and preprocessed training corpora.

Corpus	Language	Tokens		Types	
		words	words	lemmas	word MOPs
WIKIPEDIA	<i>German</i>	667.3M	9.0M	8.8M	1195
	<i>Dutch</i>	115.3M	2.0M	1.9M	908
	<i>Afrikaans</i>	12.0M	370.6K	349.5K	194

Table 18.3.: Training data statistics for multilingual compound splitting

### 18.6.3. Evaluation Format

For getting a simple way of evaluating different systems on different gold standards, a common **evaluation format** is designed that contains the following information, separated by tabs: (1) **target compound**, (2) **SPF**, (3) **LSF** and (4) **split tree** (represented as **bracketing** structure).

If there are more possible analyses (e.g., due to a variable **splitting** depth or due to alternative **constituent lemmas**), each analysis gets its own entry. The **splitting** of a **target compound** is judged as correct if there is a common **SPF** or **LSF** (depending on the evaluated discipline) between gold standard and system output in at least one evaluation format entry.

For example, splitting the compound *Fußbodenheizung* ‘underfloor heating’ with the presented recursive **MOP-based compound splitter** yields the evaluation format entries with varying **splitting** depth as shown in Figure 18.8.

Target compound	SPF	LSF	Split tree bracketing
<i>Fußbodenheizung</i>	<i>Fuß boden heizung</i>	<i>Fuß Boden Heizung</i>	<i>[[Fuß Boden] Heizung]</i>
<i>Fußbodenheizung</i>	<i>Fußboden heizung</i>	<i>Fußboden Heizung</i>	<i>[Fußboden Heizung]</i>
<i>Fußbodenheizung</i>	<i>Fußbodenheizung</i>	<i>Fußbodenheizung</i>	<i>Fußbodenheizung</i>

Figure 18.8.: Example of a **compound split** in the evaluation format

#### 18.6.4. Gold Standards

For the **intrinsic evaluation**, there is need for a gold standard containing both the gold **split points** and the underlying **constituent lemmas**. For *German*, three gold standards have been used: (1) the binary-split **nominal compound** set developed for GermaNet<sup>9</sup> by Henrich and Hinrichs (2011), subsequently referred to as HH2011GS, (2) the *N*-ary-split **nominal compound** set developed by Marek (2006) from the German newspaper magazine c't<sup>10</sup>, subsequently referred to as M2006GS and (3) the *N*-ary-split **nominal compound** set developed by Holz and Biemann (2008), subsequently referred to as HB2008GS. For *Dutch* and *Afrikaans*, we used the gold standards developed by Verhoeven et al. (2014), subsequently referred to as VZDH2014GS/NL and VZDH2014GS/AF (referring to both as VZDH2014GS).

##### HH2011GS (Henrich and Hinrichs, 2011)

**Binary splits** A key feature of this gold standard is that it provides only one **binary split** per **compound**, i.e., each **compound** is composed of exactly two **constituents**. The main problem resulting from this restriction is that the gold standard also contains **nominal compounds** being composed of more than two **atomic constituents**. For example, the **compound** *Fernsprechansagedienst* ‘telephone announcement service’ is composed of four **atomic constituents**: *fern* ‘remote’, *sprechen* ‘to speak’, *ansagen* ‘to announce’ and *Dienst* ‘service’, but it is only analyzed with the **binary split** *Fernsprech | ansagedienst*. As a consequence, a **compound splitting** result which correctly splits *Fernsprechansagedienst* into four **constituents** would be assessed as false. Since the gold standard does not exhaustively include all complex **constituents** (e.g., while *Ansage|dienst* ‘announcement service’ is included in this gold standard, there is no **split** for *fernsprechen* ‘to telephone’), it is not possible to create a recursive closure, i.e., to add **split points** to gold **constituents** if in turn these **constituents** are also **split** within the gold standard.

<sup>9</sup>[sfs.uni-tuebingen.de/GermaNet](http://sfs.uni-tuebingen.de/GermaNet)

<sup>10</sup><http://heise.de/ct>

Therefore, the [split tree](#) output is pruned to  $k_i = 2$  [constituents](#) using the tree pruning method described in [Section 18.5.1](#).

**Phrasal modifiers** Some of the [compounds](#) in HH2011GS are [phrasal compounds](#), having a phrasal [modifier](#) (see [Section 3.7.3](#)). As gold [constituent lemma](#) for the [modifier](#), a phrase with morpho-syntactic adjustments is used. For example, *Einparteiensystem* ‘one-party system’ provides the [modifier](#) phrase *eine Partei*, i.e., the first part is declined with female gender. This gender knowledge is not available for most [compound splitters](#). Moreover, the declination is not consistent. For example, while the [compound](#) *Langzeitarbeitslosigkeit* ‘long-term unemployment’ has the [modifier lemma](#) *lange Zeit*, the similar [compound](#) *Langzeitarbeitsloser* ‘long-term unemployed’ does not adjust to the female gender of *Zeit* ‘time’, i.e., the [modifier](#) phrase is *lang Zeit*. As a consequence, we decided to exclude [compounds](#) with a phrasal modifier from HH2011GS. For this purpose, all gold standard entries whose [modifier](#) contains a hyphen or space are removed from the dataset, yielding a total set of **54,148** [binary split](#) samples.

**Creation of the SPF** The initial version of HH2011GS does not have any [SPFs](#), but provides only the [target compound](#) and the [binary LSF](#) (e.g., ‘*Hühnerfleisch Huhn Fleisch*’ for ‘chicken meat’). For transforming HH2011GS into the evaluation format ([18.6.3](#)), the [SPFs](#) needs to be compiled (see [Section C.4.1](#)).

**Word MOPs vs. Gold-constituent MOPs** As described in [Section 17.2.2](#), we can compile [gold-constituent MOPs](#) from a gold standard using the gold [constituent forms](#) and the related gold [constituent lemmas](#). In [Table 18.4](#), the set of [gold-constituent MOPs](#) derived from HH2011GS is compared to the set of [word MOPs](#) (listed in [Table 18.3](#)).

	<a href="#">word MOPs</a>	<a href="#">word MOPs</a>
<a href="#">MOPs</a> in HH2011GS	54	82 (26)
<a href="#">MOPs</a> in HH2011GS	1141	—

Table 18.4.: Overlap between German Word MOPs and Gold-constituent MOPs in HH2011GS

It turns out that both the majority of [word MOPs](#) is not included in the set of [gold-constituent MOPs](#) and the majority of [gold-constituent MOPs](#) is not included among the [word MOPs](#). The latter observation is surprising. When inspecting the [gold-constituent](#)

MOPs not occurring as **word inflection**, we see that the majority ( $\sim 68\%$ ) of the exclusive **gold-constituent MOPs** are due to false annotations in the gold standard. The most frequent disagreement in the annotations has been cases where a deverbal nominal noun **modifier** is annotated with the verbal **constituent lemma**. For example, the **compound** *Abschussvorrichtung* ‘firing mechanism’ is annotated with the **modifier lemma** *abschießen* ‘to fire’, leading to the **gold-constituent MOP**  $\text{ie\ss en}\$/\text{uss}\$, which is an invalid operation for **word inflection** (while being valid for word derivation). In fact, when disregarding the **gold-constituent MOPs** extracted due to controversially annotated gold **compounds** (i.e.,  $82 \cdot (1 - 0.68) \approx 26$ ), the majority of **constituent inflection** operations is covered by **word inflection**.$

However, actually, there are some true operations in **constituent inflection** not covered by **word inflection**. The most frequent operation is the addition of the suffix *o*, which is applied to Greek or Latin roots (see also Bretschneider and Zillner (2015) for the medical domain). For example, the **compound** *Psycholinguistik* ‘psycholinguistics’ is annotated with the **modifier lemma** *Psych*, leading to the **gold-constituent MOP**  $\$/\text{o}\$, which is not found in regular **word inflection**.$

### M2006GS (Marek, 2006)

Besides **binary compound splits**, M2006GS also contains *N*-ary **compound splits** (with  $N \geq 3$ ). However, the **target compounds** in M2006GS are not **lemmatized** and occur with **head** suffixes indicating **word inflection** (e.g., pluralization or case-marking).

**Compilation** This gold standard was extracted from a corpus of the German computer magazine *c’t*<sup>11</sup> containing 15M **tokens**. After filtering **function words** and non-compounds, the remaining list of **compounds** is **split** with a rudimentary **splitter** and post-processed by human annotators in the case of low confidence. The resulting set comprises 158,657 half-automatically **split compounds**, where less than 3% might contain erroneous splits (Marek, 2006, p. 17).

**Deep Compound Splits** While this gold standard does not have the problem of a too shallow **splitting** depth (as with the **binary compound splits** in HH2011GS), there are cases which even tend to oversplit with respect to practicability. For example, there are five analyzed **constituents** for the **compound** *Daten|bank|verwaltungs|werk|zeug* ‘database administration tool’, although there is few motivation to **split** *werkzeug* ‘tool’ (e.g.,

<sup>11</sup><http://www.heise.de/ct/>

there is no need for **splitting** this **constituent** prior to **SMT**). Since most corpus-based **compound splitters** do not **split** a **word** into parts if the **word**'s corpus frequency is higher than the combination of the parts' corpus frequency, performing corpus-based **compound splitting** on this gold standard will lead to many cases of assessed **undersplitting**. An ideal way of solving the controversial issue of **splitting** depth would be the exhaustive **splitting** depth of this gold standard but presented as **bracketing** structure for determining the gold **SPF** and **LSF** for any number of **constituents** that are necessary for the task at hand.

**Prepositional Constituent Filter** Some of the entries of this gold standard contain analyses with prepositions or verb particles as **constituents**, e.g., *Vor|stellung* ‘presentation’. Since we do not consider such **targets** as **compounds** (cf. Chapter 4), these analyses are no **compound splits** in a traditional sense and thus removed. The final set contains **139,081** **compound split** samples with two or more **constituents**. Table 18.5 shows the distribution of the number of gold **constituents** provided in M2006GS.

<b>Compound size</b>	<b>Distribution</b>
2	116,218 (83.6%)
3	21,651 (15.6%)
4	1175 (0.84%)
5	35 (0.03%)
6	2 (0.001%)
total	139,081

Table 18.5.: Distribution of the number of gold **constituents** used in M2006GS

**Morphological Parse** The format of this gold standard provides information about additive and subtractive operations related to both **constituent inflection** and **word inflection**. For example, for the genitive-inflected **compound** *Suppentellers* ‘soup plate’, M2006GS provides the following parse: `suppe|n{n}+teller(s){n,v}`, where **linking elements** are separated by | and **word inflection** suffixes are written in parentheses. All possible **PoS** are written in curly brackets.

**Verbal Constituents** One issue of this gold standard is that there are often nouns for which a verbal **PoS** is presented. While the verbal interpretation of some **modifiers** are plausible (e.g., the modifier *Tanz* in *Tanzpartner* ‘dancing partner’ with the

morphological parse  $\text{tanz}\{N,V\}+\text{partner}\{N\}$  could refer to *tanzen* ‘to dance’ or *Tanz* ‘dance’), it is implausible for some other **modifiers** (e.g., the modifier *Stift* ‘pen’ in the compound *Stiftgröße* ‘pen size’ having the morphological parse  $\text{stift}\{N,V\}+\text{größe}\{N\}$ ) and impossible for a **compound head** (as shown in the morphological parse above,  $\text{suppe}\{n\}+\text{teller}(s)\{n,v\}$ ). For transforming M2006GS into the evaluation format (18.6.3), all possible verbal interpretations for the **modifiers** are included, while any verbal interpretation alternatives of **heads** are discarded.

Another issue is that there is only a verb stem available but no verb **lemma** (e.g., *tanz\_* instead of *tanzen*). Thus, there is need to map the verb stems onto the verb **lemmas**, which are considered as the infinitive verb form. In German, there are two possibilities of transforming a verb stem into the infinitive form: (1) adding the suffix *en* (e.g., *tanz* → *tanzen*) or (2) adding the suffix *n* (e.g., *jammer* → *jammern* ‘to moan’). By adding the two possible suffixes to the verb stem, we try to match with verbal corpus **lemmas** and take that verb **lemma** version with the highest corpus frequency. If both versions have the same frequency or none has corpus evidence, the *en* suffix is used as default.

In an experiment with 100 verb type samples, it turned out that 99 verb stems have correctly be transformed to the corresponding verb **lemma**. The single erroneous transformation is based on no corpus evidence and the default *en* suffix<sup>12</sup>. Since all systems in comparison are using the same **LR**, this will not have any impact on the difference of performance between the systems.

To conclude, providing some unlikely verbal interpretations of **modifiers**, this gold standard is less restrictive for false **normalization** into a verb.

**Optional Head Inflection** Since not all **compound splitting** methods **normalize constituents** with respect to **word inflection** (i.e., but only with respect to **constituent inflection**), there is need for two versions of this gold standard: (1) M2006GS with the regular **LSF**, exclusively containing bare **constituent lemmas**, and (2) M2006GS with the **LSF<sub>word</sub>**, which keeps the **word-inflected compound head**, as discussed in Section 18.5.2. For example, for the pluralized **target compound** *Umweltschutzgebieten* ‘environmental protection areas’, the corresponding **LSFs** are: *Umwelt Schutz Gebiet* (**LSF**) and *Umwelt Schutz Gebieten* (**LSF<sub>word</sub>**).

**Creation of the SPF and LSF** The initial version of M2006GS does not directly provide an **SPF** or **LSF**, but a morphological parse which includes most information for compiling

<sup>12</sup>The *en* suffix is used in 81% of all cases for transforming a verb stem to the verb **lemma**.

## 18. Multilingual Compound Splitting

the necessary formats. For the example, for the **compound** *Suppentellers*, described above, the stems *suppe* and *teller* are used for the **LSF** (while *+s* is used for **LSF<sub>word</sub>**). For the **SPF** compilation, see Section C.4.2.

**Overlap with Previous Gold Standards** Table 18.6 shows the overlap of M2006GS with HH2011GS. Both gold standards have only 16,252 **compounds** in common and their majorities are not covered by the other gold standard.

	HH2011GS	$\overline{\text{HH2011GS}}$
M2006GS	16,252	122,829
$\overline{\text{M2006GS}}$	37,900	—

Table 18.6.: Splitting gold standard overlap between HH2011GS and M2006GS

### HB2008GS (Holz and Biemann, 2008)

HB2008GS lists 700 **nominal compounds** with their **split points**. Since this gold standard does not provide any **constituent lemmas**, it is not possible to evaluate **constituent normalization** on it. Table 18.7 shows the distribution of the number of gold **constituents** provided in HB2008GS.

<b>Compound size</b>	<b>Distribution</b>
2	637 (91%)
3	50 (7.1%)
1	13 (1.9%)
total	700

Table 18.7.: Distribution of the number of gold **constituents** used in HB2008GS

**Filtering of Atomic Words** As described in Section 18.5, we decided to use a flexible tree pruning, which adapts to the **splitting** granularity of the underlying gold standard. As a consequence, the 13 **atomic words**, shown in Table 18.7, will always be predicted correctly (for tree pruning to  $k_i = 1$ ) and are therefore removed from the gold standard. The final set contains **687 compound splits**.



**Overlap with Previous Gold Standards** Table 18.8 shows the overlap of HB2008GS with the union of HH2011GS and M2006GS. All three gold standards have 306 **compounds** in common and the majority of HB2008GS, a set of 381 **compounds**, is not covered by the other two gold standards. Thus, we decided to additionally evaluate the discipline of **split point** determination on this smaller gold standard.

	HH2011GS M2006GS	$\overline{\text{HH2011GS}}$ $\overline{\text{M2006GS}}$
HB2008GS	306	381
$\overline{\text{HB2008GS}}$	176,675	—

Table 18.8.: Splitting gold standard overlap between HH2011GS+M2006GS and HB2008GS

### VZDH2014GS (Verhoeven et al., 2014)

**Initial Format** The initial format of the **compound splits** in VZDH2014GS consists of a list of **word** stems, annotated with canonical **linking elements**. For example, the Dutch **compound** *aanvangstijd* ‘start time’ is represented as ‘aanvang \_ s + tijd’, where the \_ s is the **linking element** appended to the modifier *aanvang* ‘start’.

In some cases, the **word** stem contains a morpho-phonological adaptation for being involved as **compound modifier**. For example, for the Dutch **compound** *bloem|bollen|veld* ‘flower bulb field’, the gold standard entry is *bloem + boll \_ en + veld*. The **word** stem *boll* is an **allomorph** of the **lemma** *bol* ‘bulb’. Cases of **allomorphs** have only been observed for the Dutch gold standard.

For transforming the initial format into the evaluation format, including the **SPF** and **LSF** or **LSF<sub>word</sub>** (18.6.3), the **word** stems are used as **constituent lemmas** and the concatenations of **word** stem and **linking element** as **constituent forms**.

**Number of Samples** The gold standards VZDH2014GS/NL and VZDH2014GS/AF comprise **21,941** Dutch samples and **17,362** Afrikaans<sup>13</sup> samples. Besides **binary** splits, VZDH2014GS also contains splits into three and more **constituents**. Table 18.9 shows the distribution of the number of gold **constituents** provided in VZDH2014GS.

<sup>13</sup>Six cases of Afrikaans non-compounds have been removed, because the flexible tree pruning, discussed in Section 18.5.1, would always yield the correct analysis.

Compound size	Dutch distribution	Afrikaans distribution
2	20,476 (93.3%)	14,845 (85.5%)
3	1416 (6.5%)	2349 (13.5%)
4	47 (0.21%)	156 (0.9%)
5	2 (0.009%)	12 (0.07%)
total	21,941	17,362

Table 18.9.: Distribution of the number of gold [constituents](#) used in VZDH2014GS

**Word MOPs vs. Gold-constituent MOPs** In analogy to HH2011GS, we compared the amount and share of [gold-constituent MOPs](#) and [word MOPs](#) for Dutch and Afrikaans. Table 18.10 shows the numbers for Dutch. Here, all [gold-constituent MOPs](#) also occur as [word MOPs](#). However, there are many [word MOPs](#) that do not occur as [linking element](#) in VZDH2014GS/NL (for which we conclude that these are no [constituent inflection](#) operations).

	<a href="#">word MOPs</a>	<a href="#">word MOPs</a>
<a href="#">MOPs</a> in VZDH2014GS/NL	6	0
<a href="#">MOPs</a> in VZDH2014GS/NL	902	—

Table 18.10.: Overlap between German [word MOPs](#) and [gold-constituent MOPs](#) in VZDH2014GS/NL

For Afrikaans (Table 18.11), we find a similar result. Almost all [gold-constituent MOPs](#) are among the observed [word MOPs](#). There are two [MOPs](#) which exclusively occur among the [gold-constituent MOPs](#): [\\$/slaag\\$](#) and [\\$/sorg\\$](#). These result from [lemmas](#) that have been falsely annotated as [linking elements](#) (instead of as [constituents](#)). Thus, in fact, there are no Afrikaans [constituent inflection](#) operations not occurring as [word inflection](#) operations.

	<a href="#">word MOPs</a>	<a href="#">word MOPs</a>
<a href="#">MOPs</a> in VZDH2014GS/AF	8	2 (0)
<a href="#">MOPs</a> in VZDH2014GS/AF	186	—

Table 18.11.: Overlap between German [word MOPs](#) and [gold-constituent MOPs](#) in VZDH2014GS/AF

## Other Gold Standards

There are some other gold standards developed in previous work on [compound splitting](#) that we did not use in our experiments. These are discussed in Appendix D.

### 18.6.5. Intrinsic Evaluation Method

#### Measurements

As discussed in Section 15.1.1, [intrinsic evaluation](#) measurements utilized in previous work on [compound splitting](#) have the problem of capturing the correctness of either the predicted [constituent forms](#) or the predicted [constituent lemmas](#), but not both.

The most influential evaluation measure is presented by Koehn and Knight (2003). They propose five **evaluation categories**:

- (a) **correct split** (cs): [words](#) that should be [split](#) and were [split](#) correctly  
e.g., *Hundehütte* ‘doghouse’ → *Hunde* | *hütte* ‘dog | house’ ✓
- (b) **correct not** (cn): [words](#) that should not be [split](#) and were not  
e.g., *Fenster* ‘window’ → *Fenster* ✓
- (c) **wrong not** (wn): [words](#) that should be [split](#) but were not  
e.g., *Fensterscheibe* ‘windowpane’ → *Fensterscheibe* ✗
- (d) **wrong faulty split** (wf): [words](#) that should be [split](#), were [split](#), but wrongly (either false [split point](#), too many or too few [split points](#))  
e.g., *Kreditkartenanbieter* ‘credit card provider’ → *Kreditkar|tenanbieter* ✗  
→ *Kredit|karten|an|bieter* ✗  
→ *Kreditkarten|anbieter* ✗
- (e) **wrong split** (ws): [words](#) that should not be [split](#), but were  
e.g., *Fenster* ‘window’ → *Fen|ster* ✗

where category ‘(c) **wrong not**’ measures the degree of [undersplitting](#), category ‘(d) **wrong faulty split**’ the degree of [false splitting](#) and category ‘(e) **wrong split**’ the degree of [oversplitting](#).

Based on these five evaluation categories, Koehn and Knight (2003) propose the three **evaluation measurements** precision (P, Formula 18.5), recall (R, Formula 18.6) and accuracy (Acc, Formula 18.7).

$$P = \frac{cs}{cs + wf + ws} \quad (18.5)$$

$$R = \frac{\text{cs}}{\text{cs} + \text{wn} + \text{wf}} \quad (18.6)$$

$$Acc = \frac{\text{cs} + \text{cn}}{\text{cs} + \text{cn} + \text{wn} + \text{wf} + \text{ws}} \quad (18.7)$$

For gold standards that contain only true **compounds** (i.e., all gold standards described in Section 18.6.4), there are no categories **cn** and **ws**. As a result, recall equals accuracy ( $P \hat{=} Acc$ ) and precision is always as high as recall or higher ( $P \geq R$ ).

Additionally, as a combination of precision and recall, the harmonic mean (F<sub>1</sub>-Score) is used as shown in Formula 18.8.

$$\text{F}_1\text{-Score} = \frac{2 \cdot P \cdot R}{P + R} \quad (18.8)$$

While the measurements of Koehn and Knight (2003) can be understood as either counting the correct **split points** (or **constituent forms**) or the predicted **constituent lemmas**, we propose to evaluate the **splitting** quality with respect to both disciplines:

1. Determination of the correct **split points** (SPX,  $X \in \{P, R, Acc, F_1\}$ )
2. **Normalization** of the resulting **constituent forms** (NormX,  $X \in \{P, R, Acc, F_1\}$ )

All four measurements (i.e., precision, recall, accuracy and F-score) are represented for both disciplines, SPX and NormX, e.g., ‘SPR’ refers to the recall for **split point** determination and ‘NormAcc’ measures the accuracy for the **constituent normalization**.

### Statistical Significance Test

For testing the statistical significance of the performance difference between two **compound splitting** systems (or setups), we use the **Approximate Randomization Test** (Yeh, 2000), with a significance level of  $p < 0.05$ . For the sake of simplicity, statistical significance is only measured for the performance differences between the **word MOP**-based **compound splitting** approach and the systems in comparison, but not between the compared systems.

For comparing the performance of the **word MOP**-based approach with published performance numbers (e.g., those in Verhoeven et al. (2014)), we use the **z-test for proportions**, with a significance level of  $p < 0.05$ .

### 18.6.6. External Compound Splitting Methods in Comparison

As discussed in Section 16.3.2, we decided to compare our MOP-based compound splitter against two German linguistically motivated methods, because these provide the highest benchmarks for German compound splitting.

#### Fritzing and Fraser (2010)

**Word-inflected Heads** The SMOR- (Schmid et al., 2004) and corpus-based approach of Fritzing and Fraser (2010), outlined in Section 16.2, produces a ranked list of compound splits with different numbers of constituents, represented in an LSF with a word-inflected head, i.e., an LSF<sub>word</sub>. For example, the pluralized compound *Hungersnöten* ‘famine’ is split by the system into the LSF *Hunger Nöten*, i.e., while the modifier is normalized, the word inflection of the head is still present. For evaluating the discipline of constituent normalization, the gold standards containing inflected compounds (i.e., M2006GS) provide a version with LSF<sub>word</sub>. While the gold standard HB2008GS contains word-inflected compounds, it only provides an SPF and thus, no LSF<sub>word</sub> is necessary.

**Creation of the Split Point Format** The SPF is compiled using the algorithm presented in Appendix C. More details are given in Section C.4.3.

Subsequently, we will refer to this compound splitting approach as FF2010.

#### Weller and Heid (2012)

**Updated Version** The compound splitting method of Weller and Heid (2012), outlined in Section 16.2, has been revised in several iterations in the recent years and represents an exemplar of a corpus-based compound splitting method which is optimized for a certain language. For the experiments presented in this thesis, the most recent<sup>14</sup> version has been used. It relies on a PoS<sup>15</sup>-tagged and lemmatized training corpus. For avoiding noisy splits, the training corpus is filtered. Therefore, a hand-crafted list of valid German character bigrams is used. Words having a length between 3 and 5 characters are filtered using a bilingual dictionary<sup>16</sup>.

---

<sup>14</sup>Release date: 13<sup>th</sup> October 2016

<sup>15</sup>Three possible PoS tags are considered: NN (for nouns and named entities), V (for any full verb) and ADJ (for adjectives, adverbs and some prepositions)

<sup>16</sup>Marion Weller-Di Marco provided us with a prefiltered version of the English-to-German dict.cc.

**System Output** The corpus-based approach of Weller and Heid (2012), including an extensive list of morphological rules for modelling [constituent inflection](#), produces a ranked list of  $N$ -ary [compound splits](#) with both [lemmatized](#) and [word-inflected heads](#), i.e., with [LSF](#) and [LSF<sub>word</sub>](#). All [constituents](#) are annotated with a [PoS](#). Figure 18.9 shows an excerpt of the ranking output of the system of Weller and Heid (2012) for the pluralized [compound](#) *Abenteuerromane* ‘adventure novels’ and the [compound](#) *Hühnerfutter* ‘chicken feed’.

Target compound	PoS-tagged LSF	PoS-tagged LSF <sub>word</sub>
abenteuerromane	abenteuer_NN roman_NN	abenteuer_NN romane_NN
abenteuerromane	abenteuerroman_NN	abenteuerromane_NN
abenteuerromane	abenteuern_V roman_NN	abenteuern_V romane_NN
abenteuerromane	abene_NN teuer_NN roman_NN	abene_NN teuer_NN romane_NN
abenteuerromane	aben_NN teuer_NN roman_NN	aben_NN teuer_NN romane_NN
abenteuerromane	abente_NN euer_NN roman_NN	abente_NN euer_NN romane_NN
hühnerfutter	huhn_NN futter_NN	huhn_NN futter_NN
hühnerfutter	hühnerfutter_NN	hühnerfutter_NN

Figure 18.9.: Examples of a ranking output produced by the splitting system of Weller and Heid (2012)

For evaluating the system of Weller and Heid (2012), we only consider the [LSF](#) in the second column and disregard the [LSF<sub>word</sub>](#) as well as any [PoS](#) tags.

**Creation of the Split Point Format** As illustrated for the output of *Hühnerfutter* in Figure 18.9, the system of Weller and Heid (2012) does not produce an [SPF](#). Therefore, we apply the linear [SPF](#) compilation (outlined in Algorithm C.1) to the [constituent lemmas](#).

Subsequently, we will refer to this [compound splitting](#) approach as WH2012.

## Verhoeven et al. (2014)

The system of Verhoeven et al. (2014) is not publicly available, but since it has the same gold standard, it is possible to compare the [MOP](#)-based [compound splitting](#) approach presented in this thesis against the **performance numbers** presented in Verhoeven et al. (2014). The authors provide only accuracy numbers for determining the correct [split point](#) in *Dutch* and *Afrikaans*. Thus, the [word MOP](#)-based [compound splitter](#) can only be compared to these performance numbers using the SP $Acc$  measurement.

Subsequently, we will refer to this [compound splitting](#) approach as VZDH2014.

### 18.6.7. Results and Discussion

In this section, we present various **evaluation blocks** for illustrating the **splitting** quality of the proposed **splitting** method and in order to find information for answering the **research questions** posed in Section 15.2. The research questions will be explicitly answered in the conclusion section (21.2) of bottom line (Chapter 21) of the **compound splitting** part.

#### Compound Splitting Features

**Setup** In the previous sections, we presented and motivated several features which have been used for **compound splitting**. In this evaluation block, all features are evaluated using the **word MOP**-based **compound splitter** applied to the gold standard M2006GS. We use this gold standard, because it includes **word-inflected compounds**. The default setup (DEFAULT) uses all proposed features. For each feature, we show the performance number for the **compound splitter** that has all features enabled except for the discussed feature. The evaluated features are:

1. **Constituent content word** restriction (as discussed in the beginning of Chapter 18)
2. **Head** EQuality (*hEQ*) restriction on the **PoS** of **head** and **compound** ((Formula 18.4), as discussed in Section 18.3)
3. Prior **MOP lemmatization**, discussed in Section 18.4.1
4. **Compound content word** restriction, discussed in Section 18.4.2
5. **PoS** agreement restriction for the **modifiers** (Section 18.4.3)
6. **Lexeme** agreement restriction for the **head** (Section 18.4.4)
7. Different means used in Formula 18.3: arithmetic mean vs. geometric mean vs. harmonic mean

Table 18.12 shows the evaluation of all proposed **compound splitting** features.

**Results** For almost all features, there is a statistically significant<sup>17</sup> (♥) deterioration in performance when subtracting it from the DEFAULT setup for at least some metrics.

---

<sup>17</sup>Approximate Randomization Test (Yeh, 2000),  $p < 0.05$

## 18. Multilingual Compound Splitting

Feature	SPX				NormX			
	P	R	Acc	$F_1$	P	R	Acc	$F_1$
DEFAULT	97.3	96.4	96.8	96.8	87.3	86.6	87.0	87.0
⊖ Constituent content words	96.4 🚩	95.6 🚩	96.0 🚩	96.0 🚩	86.3 🚩	85.6 🚩	86.0 🚩	86.0 🚩
⊖ $hEQ$	97.4 🟢	96.8 🟢	97.1 🟢	97.1 🟢	85.6 🚩	85.0 🚩	85.3 🚩	85.3 🚩
⊖ Prior MOP lemmatization	97.0 🚩	96.3 🚩	96.7 🚩	96.7 🚩	86.1 🚩	85.5 🚩	85.8 🚩	85.8 🚩
⊖ Compound content words	97.3 🚩	96.4 🚩	96.8 🚩	96.8 🚩	87.3	86.6 🚩	87.0 🚩	87.0 🚩
⊖ Modifier MOP PoS agreement	97.2 🚩	96.4 🚩	96.8 🚩	96.8 🚩	86.2 🚩	85.5 🚩	85.9 🚩	85.9 🚩
⊖ Head MOP lexeme agreement	97.3	96.6 🟢	96.9 🟢	96.9 🟢	84.5 🚩	83.9 🚩	84.2 🚩	84.2 🚩
⊘ Arithmetic mean	88.9 🚩	88.5 🚩	88.7 🚩	88.7 🚩	79.6 🚩	79.2 🚩	79.4 🚩	79.4 🚩
⊘ Harmonic mean	96.6 🚩	94.9 🚩	95.8 🚩	95.8 🚩	86.8 🚩	85.3 🚩	86.1 🚩	86.1 🚩

Table 18.12.: Evaluation of all proposed [compound splitting](#) features; numbers in %

Removing the two features  $hEQ$  and the [lexeme](#) agreement for [head MOPs](#) yields a significant (🟢) increase in SPX, but a strong drop in NormX. Removing the [content word](#) restriction for the complete [compound](#) does not have much impact on the performance. This is to be expected, because the [compound content word](#) restrictor is motivated for running text, where [function words](#) could be [split](#), e.g., *ob|wohl* ‘although’. These non-compound [targets](#) do not occur in M2006GS.

Using the geometric mean significantly outperforms the usage of the arithmetic or harmonic mean. This is in line with the observations made by Stymne (2008).

**Conclusion** Subsequently, we will use the DEFAULT feature setup, because it provides the best trade-off between performance for finding the correct [split point](#) (SPX) and [normalizing](#) the resulting [constituent forms](#) (NormX).

### MOP Set Comparison

**Setup** In this evaluation block, we compare the [splitting](#) performance when using [word MOPs](#) with the [splitting](#) performance of the other types of MOP sources, described in Section 17.2. Tables 18.13 and 18.14 show the performance with respect to the SPX and NormX metrics for all three German gold standards and the Dutch/Afrikaans gold standards when using the various MOP sets: [word](#) ([word MOPs](#), as described in Section 17.2.1), HH2011GS, M2006GS, VZDH2014GS/NL and VZDH2014GS/AF ([gold-constituent MOPs](#), as described in Section 17.2.2), Langer ([hand-crafted constituent MOPs](#), as described in Section 17.2.3) and null (i.e., the set containing only the [null-MOP](#) with the



frequency of the corresponding [word MOP](#)).

For HB2008GS, there are no [gold-constituent MOPs](#), because this gold standard only provides [split points](#) (i.e., [constituent forms](#)) but no [constituent lemmas](#), from where the [gold-constituent MOPs](#) could be compiled. The learned [gold-constituent MOPs](#) are divided into [MOPs](#) derived from gold [modifiers](#) and into [MOPs](#) derived from gold [heads](#). For [gold-constituent MOPs](#), no agreement restrictions with respect to [PoS](#) or [lexeme](#) (outlined in Section 18.4.3 and Section 18.4.4) are used. In particular, the [PoS](#) agreement cannot be applied, because there is usually no [PoS](#) information provided in a [compound splitting](#) gold standard. For the [head constituent](#), all learned [gold-constituent MOPs](#) are derived from the [word-inflected compound head](#). As this source of [MOPs](#) is quite restricted, we expect that there is no need for the [lexeme](#) agreement restrictor (designed for [word MOPs](#)). For the Afrikaans gold standard, for which [word inflection](#) is retained, there are no learned [MOPs](#) for the [head](#). Thus, we use [word MOPs](#) and the [lexeme](#) agreement restrictor for the [heads](#) in VZDH2014GS/AF.

The [hand-crafted constituent MOPs](#) derived from Langer (1998) (see Appendix B) are based on the knowledge that only the [modifiers](#) undergo [constituent inflection](#), whereas the [head](#) undergoes [word inflection](#). Thus, for this [MOP](#) set, the [PoS](#) agreement restriction on the [modifier](#) is not used. For determining the [head lemmas](#) the [word MOPs](#) in combination with the [lexeme](#) agreement restriction is used.

The [null-MOP](#) is used for [normalizing modifiers](#), because this baseline is intended to show indirectly the degree of non-trivial [constituent inflection](#) (i.e., with not-null operations) in a language. Since this baseline would be artificially bad for [word-inflected](#) (e.g., pluralized) [compounds](#), in analogy to the [hand-crafted constituent MOPs](#), the [heads](#) are [normalized](#) using [word MOPs](#) in combination with the [lexeme](#) agreement restriction.

## Results and Discussion for German Comparison

**Word MOPs** As expected, the results show that the performance of [compound splitting](#) using [word MOPs](#) is solid, but it is clearly outperformed by the [gold-constituent MOPs](#) and the [hand-crafted constituent MOPs](#) for some metrics. For all three gold standards, [word MOPs](#) show similar performance for determining the correct [split points](#) (SPX). While there are no significant differences in SPX for the smaller gold standard HB2008GS and while the [word MOPs](#) seem to have comparable SPX numbers for HH2011GS, for the largest gold standard, M2006GS, the [word MOPs](#) are significantly (👍)

## 18. Multilingual Compound Splitting

Gold Standard	MOP Set	SPX				NormX			
		<i>P</i>	<i>R</i>	<i>Acc</i>	<i>F</i> <sub>1</sub>	<i>P</i>	<i>R</i>	<i>Acc</i>	<i>F</i> <sub>1</sub>
HH2011GS	word	98.1	96.9	97.5	88.0	87.0	87.5		
	HH2011GS	98.0	96.8 🍷	97.4	90.3 🍷	89.1 🍷	89.7 🍷		
	Langer	98.4 🍷	96.9	97.6 🍷	90.1 🍷	88.7 🍷	89.4 🍷		
	null	95.2 🍷	77.3 🍷	85.4 🍷	73.3 🍷	59.5 🍷	65.7 🍷		
M2006GS	word	97.3	96.4	96.8	87.3	86.6	87.0		
	M2006GS	97.4 🍷	96.7 🍷	97.1 🍷	88.8 🍷	88.1 🍷	88.4 🍷		
	Langer	97.5 🍷	96.5	97.0 🍷	89.4 🍷	88.5 🍷	88.9 🍷		
	null	90.3 🍷	76.2 🍷	82.7 🍷	64.5 🍷	54.4 🍷	59.0 🍷		
HB2008GS	word	98.7	97.1	97.9	not evaluated				
	Langer	98.7	96.2	97.4					
	null	94.1 🍷	74.4 🍷	83.1 🍷					

Table 18.13.: Results for German [compound splitting](#) - MOP set comparison; numbers in %

outperformed by the [gold-constituent MOPs](#) and the [hand-crafted constituent MOPs](#).

For the task of [normalizing the constituent forms](#), the [word MOP](#)-based approach is always significantly (🍷) outperformed (by about 1-2 percentage points) by the [gold-constituent MOPs](#) and the [hand-crafted constituent MOPs](#).

One reason for this is the open issue for finding a trade-off between [PoS](#) agreement and [lexeme](#) agreement restrictions for the [modifiers](#), outlined in Section 18.4.5: the [PoS](#) agreement is too lenient and allows invalid [normalization](#) to high-frequent corpus [lemmas](#). This does not happen for [hand-crafted constituent MOPs](#) that do not model operations that are exclusively used in [word inflection](#). For example, while the [compound Stangenwaffe](#) ‘pole weapon’ is correctly split into the [constituent forms](#) *Stangen* and *waffe* by both [MOP](#) sets, the [word MOPs](#) falsely [normalize](#) the [modifier](#) to the verb *stehen* ‘to stand’, whereas the [hand-crafted constituent MOPs](#) correctly normalize it to *Stange* ‘pole’. In this case, the misleading [word MOP](#) is `eh/ang`, which is usually used for transforming prefix verbs with the verb stem *gehen* ‘to go’ into the participle form, e.g., *umgehen* → *umgangen* ‘to circumvent’. Another example of [word MOP](#)-induced noise is the [compound System|still|stand](#) ‘system stop’ (lit. ‘system inactive state’), where the second [constituent form](#), *still*, is falsely normalized to *stellen* ‘to set’ using the [word MOP](#) `e/i:en$/` (which is used for the imperative form of some German verbs, e.g., *versprechen* → *versprich* ‘to promise’). Since this operation of [word inflection](#) is never observed (for [modifiers](#)) in a [compound splitting](#) gold standard, this false [normalization](#)

does not happen, when using **gold-constituent MOPs**. On the other hand, the set of **hand-crafted constituent MOPs** derived from Langer (1998) is not exhaustive and lacks several important operations, e.g., the truncation of  $n$ , which frequently happens to verbal **modifiers**. For example, while the **compound** *Förder|mittel* ‘development funds’ is correctly normalized to the **modifier lemma** *fördern* ‘to promote’ by using the **word MOP**  $n\$/\$,$  the only possible **hand-crafted constituent MOP** from Langer (1998) is  $o/\ddot{o}:\$/er\$,$  leading to the named entity **modifier** *Ford*. Although this **MOP** is also included in the set of **word MOPs**, the **PoS** agreement restriction avoids **constituent inflection** on named entities<sup>18</sup>.

**Gold and Hand-crafted Constituent MOPs** The approach of **gold-constituent MOPs**, which can be considered as an **UPPER bound** with respect to the **MOP** quality, is inferior to the approach of using **hand-crafted constituent MOPs**. One reason for this is the fact that we are using the **lexeme** agreement restrictor for **hand-crafted constituent MOPs** but not for **gold-constituent MOPs**, as discussed above in the setup of this evaluation block. Actually, the lack of **lexeme** agreement produces **splitting** analyses with falsely normalized **heads**. For example, in the **compound** *Textil|techniker* ‘textile engineer’, the **head** is falsely **normalized** to *Technik* ‘engineering’ using the **gold-constituent MOP**  $\$/er\$.$  While this **MOP** is valid for pluralized **compound heads** as in *Spiegel|bilder* ‘mirror images’, it is not applicable to the **head lemma** *Technik*. Using a **lexeme** agreement restriction, this analysis would be discarded.

**The Null-MOP** **Compound splitting** using the **null-MOP** baseline for modeling **constituent inflection** on the **modifier** underperforms heavily (♥) for all three gold standards. However, precision for both **split point** determination (*SP*) and **constituent normalization** (*NormP*) achieves more than 90%. This shows that the main problem of using the **null-MOP** is **undersplitting**, i.e., the lack of operations for **constituent inflection** avoids finding any possible **constituent lemma** for any potential **constituent form**. This baseline illustrates that the usage of knowledge about **constituent inflection** is indispensable for **compound splitting**, in particular for the task of **constituent normalization**. Using only the **null-MOP** can only provide a correct **splitting** for **compounds** where all **constituent forms** are equal to their corresponding **constituent lemmas**. While there is a higher chance for finding such **compounds** among the **binary compound splits** of HH2011GS for the gold standards also providing **compounds** with three or more **constituents**, the

<sup>18</sup>This holds for a language’s **PoS** tag set in which named entities have a unique **PoS**.

chance of non-constituent-inflected compounds decreases. As a result, this baseline’s performance is best for HH2011GS. The degree of non-trivial constituent inflection operations cannot be measured directly using the performance numbers of the null-MOP baseline, because a false compound split does not necessarily mean a compound with non-trivial operations: an alternative source of error is an unlikely but possible split point selection (e.g., *Eidotter* ‘egg yolk’ split into *Eid* | *otter* ‘oath otter’), as will be discussed in Chapter 19.

**Differences across the Gold Standards** For all systems, there is a drop in performance when switching from the gold standard HH2011GS to M2006GS. This is to be expected, because besides binary compound splits, M2006GS also includes  $N$ -ary splits for  $N \geq 3$ . Compound splitting into more than two constituents is a more challenging task. Moreover, while the compounds in HH2011GS are lemmatized, M2006GS also includes word-inflected (e.g., pluralized or case-marked) compounds. For lemmatized compounds, the high-frequent null-MOP can map the head form onto the head lemma, whereas for word-inflected compounds, the correct MOP for lemmatizing the head is necessary.

Gold Standard	MOP Set	SPX				NormX			
		$P$	$R$	$Acc$	$F_1$	$P$	$R$	$Acc$	$F_1$
DUTCH									
VZDH2014GS/NL	word	96.2	94.6	95.4	81.0	79.7	80.4		
	VZDH2014GS/NL	97.0	91.8	94.3	93.0	88.1	90.5		
	null	93.9	81.9	87.5	74.3	64.8	69.2		
AFRIKAANS									
VZDH2014GS/AF	word	93.1	82.9	87.7	87.0	77.4	81.9		
	VZDH2014GS/AF	94.0	82.1	87.6	90.8	79.3	84.6		
	null	89.5	63.4	74.3	82.2	58.2	68.1		

Table 18.14.: Results for Dutch and Afrikaans compound splitting - MOP set comparison; numbers in %

**Results and Discussion for Dutch and Afrikaans Comparison** Table 18.14 shows the results for the two gold standards of Verhoeven et al. (2014), the Dutch VZDH2014GS/NL and the Afrikaans VZDH2014GS/AF.

**Split Point Determination** The first result is that the word MOP-based approach show a solid performance for both *Dutch* and *Afrikaans*. For determining the correct

split points, the approach based on word MOPs is significantly (👍) outperformed by the gold-constituent MOPs in SPP, but in turn significantly (👎) outperforms them with respect to SPR/Acc.

**Constituent Normalization** In analogy to the German compound splitting, the performance of normalizing Dutch and Afrikaans constituents is the much harder task in which word MOPs are significantly inferior to gold-constituent MOPs. Comparing the similar languages *Dutch* and *Afrikaans*, it is interesting to see that the word MOP-based normalization precision (NormP) is 6% better for *Afrikaans* (which performs similarly precise than *German*), for which there is only the smallest training corpus available (Table 18.3).

**Allomorphs in Dutch Gold Standard** One of the reasons for the moderate normalization precision of Dutch word MOPs are the allomorphs for some constituents in the Dutch gold standard VZDH2014GS/NL. For example, the Dutch compound *aalbessensap* ‘currant juice’ is analyzed with the word MOP-based approach as *aalbes + sap* ‘currant + juice’, whereas the gold standard demands *aalbess + sap*, where *aalbess* is an allomorph of the correct lemma *aalbes*.

In contrast, the gold-constituent MOPs are directly derived from the allomorphs in VZDH2014GS/NL. In some cases, an allomorph occurred as corpus lemma in isolation (e.g., in the case of a misspelling). In such a case, the gold-constituent MOP yield to the ‘correct’ (allomorph) lemma. For example, the Dutch compound *dievenklauw* ‘anti-lift pin’ (lit: ‘thief claw’) has the gold modifier *diev*, i.e., the corresponding gold-constituent MOP is  $\$/en\$ . Since the allomorph *diev* is observed in Dutch WIKIPEDIA (with a corpus frequency of 6), the approach using gold-constituent MOPs match with the gold modifier. In contrast, the word MOPs include the MOP  $f\$/ven\$ , which yield the more frequent corpus lemma *dief* (having a corpus frequency of 586).

**Performance for Afrikaans** Since the Afrikaans gold standard does not contain allomorphs, there is not such a big performance gap between using word MOPs and gold-constituent MOPs. However, the Afrikaans performance is worst with respect to recall/accuracy. As shown in Formula 18.6, the cause for bad recall values (and still good precision values) is the evaluation category WN. The small Afrikaans corpus size has a relevant impact on the recall/accuracy of compound splitting and thus causes undersplitting.

**Conclusion** In this evaluation block, we compared the **compound splitting** performance on different **MOP** sets: **word MOPs**, **gold-constituent MOPs**, **hand-crafted constituent MOPs** and the **null-MOP**; for three languages (*German*, *Dutch* and *Afrikaans*) and five gold standards (HH2011GS, M2006GS, HB2008GS, VZDH2014GS/NL and VZDH2014GS/AF). For the solid German gold standards, the **word MOP**-based approach is comparable to the approaches using **gold-constituent MOPs** and **hand-crafted constituent MOPs**. While the performance for Dutch and Afrikaans **compound splitting** is still acceptable, there is a clear performance drop. This has several reasons: (1) the Dutch gold standard contains **allomorphs** and no isolated corpus **lemmas** and (2) the training data (in particular the one for Afrikaans) is smaller than the German training corpus.

## External System Comparison

**Setup** In this evaluation block, we compare the performance of the **MOP**-based **compound splitter** with **compound splitting** methods presented in previous work, outlined in Section 18.6.6. A first version of the **word MOP**-based **compound splitter** presented in this thesis has been published in Ziering and Van der Plas (2016). Thus, subsequently, we will refer to the **word MOP**-based **splitting** method as ZvdP2016.

As discussed in Section 18.5.1, the deepest **split tree** produced by ZvdP2016 is iteratively pruned down to the size that matches with the number of gold **constituents**,  $k_{gold}$ . Allowing the same flexibility for the external systems in comparison, for each  $k$ -value, we collect the highest-scored **split** option from the proposed ranking output. While the deepest **split tree** produced by ZvdP2016 is the best-scored result, the ranking output of the external systems include analyses with more **constituents** than the best-scored analysis has. This gives the external systems an advantage, because lower-scored splits with larger  $k$ -values are not available for ZvdP2016.

Table 18.15 shows the results for applying the different **compound splitting** methods to the three German gold standards. In the most recent versions of each **splitting** method, all methods have full coverage, i.e., they produce an output for each **target compound** (i.e., an  $N$ -ary **compound split** or an analysis as **atomic word**). Table 18.18 compares the **split point** accuracy (SPAcc) of the **word MOP**-based approach applied to VZDH2014GS with the accuracy numbers presented in Verhoeven et al. (2014).

## Results and Discussion for German Compound Splitters

## 18. Multilingual Compound Splitting

Gold Standard	System	SPX				NormX			
		<i>P</i>	<i>R</i>	<i>Acc</i>	<i>F</i> <sub>1</sub>	<i>P</i>	<i>R</i>	<i>Acc</i>	<i>F</i> <sub>1</sub>
HH2011GS	ZvdP2016	98.1	96.9	97.5		88.0	87.0	87.5	
	FF2010	98.5👍	92.3👎	95.3👎		95.0👍	89.0👍	91.9👍	
	WH2012	98.1	96.8	97.4		91.8👍	90.7👍	91.2👍	
M2006GS	ZvdP2016	97.3	96.4	96.8		87.3	86.6	87.0	
	FF2010	92.9👎	88.2👎	90.5👎		90.9👍	86.4👎	88.6👍	
	WH2012	98.9👍	98.6👍	98.7👍		93.1👍	92.8👍	92.9👍	
HB2008GS	ZvdP2016	98.7	97.1	97.9		not evaluated			
	FF2010	96.6👎	91.4👎	93.9👎					
	WH2012	98.4	98.4	98.4					

Table 18.15.: Results for German compound splitting - external system comparison

**Split Point Determination** The first result is that the [word MOP](#)-based [compound splitting](#) approach, ZvdP2016, shows a solid performance and is partially competitive to the other German [compound splitters](#). In particular, for the discipline of determining the correct [split points](#) for the gold standards HH2011GS and HB2008GS. One advantage of ZvdP2016 over WH2012 is that WH2012 does not [split hyphenated compounds](#) whose [modifier](#) has only one character, e.g., *A-Säule* ‘A-pillar’ or *X-Achse* ‘abscissa’. Although ZvdP2016 does not allow single characters as [constituents](#), [compounds](#) containing a [split point marker](#) undergo a special treatment, as discussed in Section 18.1.1. Another limitation of WH2012 is that [compounds](#) are maximally [split](#) into four [constituents](#), by default. This means, [compounds](#) with five or more [constituents](#) cannot be [split](#) correctly by WH2012 (e.g., the [compound](#) *Funk|netz|werk|schnitt|stelle* ‘wireless network interface’). While FF2010 does not have the limitations of [compound size](#) (in terms of [atomic constituents](#)), this system, which relies on the lexicon-based SMOR, suffers from [undersplitting](#) and provides more [atomic](#) analyses for which ZvdP2016 and WH2012 provide a true [splitting](#).

**Constituent Normalization** In analogy to all results presented for the [MOP](#) set comparison, in the discipline of [constituent normalization](#), the [word MOP](#)-based approach in ZvdP2016 is significantly inferior to the language-specific [compound splitters](#) in comparison. One reason for this is similar to the observations made when comparing [word MOPs](#) to [gold-constituent MOPs](#): there are misleading [word MOPs](#) that pass the [PoS](#) agreement restrictor and yield a false high-frequent [constituent lemma](#). For example, the [modifier](#) in the [compound](#) *Marken|artikel* ‘branded article’ is falsely normalized

## 18. Multilingual Compound Splitting

to *Mark* ‘marrow’ using the [word MOP](#) `$/en$`, which is valid for nouns such as *Zahl* → *Zahlen* ‘number → numbers’ in [compounds](#) such as *Zahlen|code* ‘number code’. Another issue is the missing language-specific knowledge about the correct types of [constituent forms](#) for a given [PoS](#). For example, operations for [constituent inflection](#) of German verbs usually include the truncation to the verb stem (e.g., *essen* → *ess* ‘to eat’ as in *Ess|verhalten* ‘eating behavior’) or the *n*-truncation (e.g., *zeigen* → *zeige* ‘to show’ as in *Zeige|finger* ‘forefinger’), i.e., German verbs never undergo a [null](#) operation during [constituent inflection](#) (modeled by the [null-MOP](#)) which would lead to the full infinitive form. Lacking this knowledge, the [null-word MOP](#) leads to an infinitive verb form as [modifier](#), as in *Ehren|gast* ‘guest of honor’ [normalized](#) to *ehren + Gast* ‘to honor + guest’.

**Undersplitting** For M2006GS, WH2012 significantly (👍) outperforms ZvdP2016 in SPX. An error analysis revealed that the main problem of ZvdP2016 (with around 70% of all cases) is [undersplitting](#) for [compounds](#) with three or more [constituents](#). The [binary splitting](#) architecture of ZvdP2016, shown in Figure 18.1, stops recursion if the [binary splitter](#) ranks an [atomic](#) analysis highest. As discussed in the setup of this evaluation block, this limitation gives an advantage for external systems providing a ranked output of [compound splits](#) with different [splitting](#) depths. Actually, there are much more [compound splits](#) with three or four [constituents](#) provided by WH2012, as shown in Table 18.16.

<i>k</i>	Distribution for ZvdP2016	Distribution for FF2010	Distribution for WH2012
1	139,081	139,081	91,665
2	137,899	132,170	137,939
3	38,828	15,383	<b>96,092</b>
4	6961	538	<b>32,141</b>
5	1024	9	—
6	109	—	—
7	9	—	—

Table 18.16.: Distribution of [splitting](#) depths in M2006GS

Table 18.17 shows some examples of [compounds](#) that have been undersplit by ZvdP2016 but still have at least one correct [split point](#).

In most cases, the unsplit [constituents](#) have a very strong [semantic association](#) or are non-compositional [compounds](#) (Section 3.8.1). This is also illustrated by the fact that there are asymmetric translations of the unsplit [constituents](#) in English, e.g., *Gastgeber*



Target compound	Splitting of ZvdP2016	Splitting of WH2012
<i>Gastgebernetzes</i> ‘host network’	<i>Gastgeber netzes</i>	<i>Gast geber netzes</i>
<i>Leinwandgröße</i> ‘canvas size’	<i>Leinwand größe</i>	<i>Lein wand größe</i>
<i>Multimediamesse</i> ‘multimedia fair’	<i>Multimedia messe</i>	<i>Multi media messe</i>
<i>Fotowerkzeuge</i> ‘photo tools’	<i>Foto werkzeuge</i>	<i>Foto werk zeuge</i>
<i>Hintergrundprozessen</i> ‘background process’	<i>Hintergrund prozessen</i>	<i>Hinter grund prozessen</i>
<i>Weltkriegsszenarium</i> ‘world war scenario’	<i>Weltkriegs szenarium</i>	<i>Welt kriegs szenarium</i>

Table 18.17.: Examples of different splitting depths for ZvdP2016 and WH2012

‘host’, *Leinwand* ‘canvas’ or *Werkzeug* ‘tool’. For FF2010, the trend of **undersplitting** is even stronger. For example, the **compound** *Bildschirmarbeitsplätze* ‘screen workplaces’ is analyzed with the **SPF** having four **constituent forms** *Bild|schirm|arbeits|plätze* by ZvdP2016 and WH2012, but with the **SPF** having only two **constituent forms** *Bildschirm|arbeitsplätze* by FF2010.

As a consequence, we argue for not treating partial **undersplitting** (i.e., with partially correct **split points**) the same as full **undersplitting** or totally false **splitting** (i.e., hitting no correct **split point** at all). This could be achieved by switching from the **word**-level to the **split point**-level for evaluating **compound splitting**. This will be addressed in future work.

We neglect the option of forcing a **split** (i.e., discarding an **atomic** analysis as long as possible) in the recursive **splitting** architecture of ZvdP2016 in order to be more competitive with WH2012 in the current setup, because this would lead to an unrealistic and impractical **splitting** policy, which is not applicable to an **NLP** task such as **RTE** (which will be discussed in Chapter 20).

Gold standard	System	SP Acc
VZDH2014GS/NL	ZvdP2016	94.6
	VZDH2014	91.5 🇷🇺
VZDH2014GS/AF	ZvdP2016	82.9
	VZDH2014	88.3 🇺🇸

Table 18.18.: Results for Dutch/Afrikaans compound splitting - external system comparison

## Results and Discussion for Dutch and Afrikaans Compound Splitters

**Split Point Accuracy** Since Verhoeven et al. (2014) only presents accuracy numbers for determining the correct **split point**, Table 18.18 compares ZvdP2016 only in the **split point** accuracy (*SPAcc*). The first result is that ZvdP2016 outperforms VZDH2014 significantly (👉) in *Dutch*. In contrast, the original system of Verhoeven et al. (2014) is significantly superior (👍) in **splitting** Afrikaans **compounds**.

**Error Analysis for Afrikaans** Although there is no possibility to compare the samples that are processed correctly by VZDH2014 and not by ZvdP2016, because only performance numbers are available, a closer look into the false **compound splits** of ZvdP2016 reveal that there are 1914 cases of the evaluation category *wrong not* (**wn**) and only 1059 cases of the category *wrong faulty split* (**wf**). This means that the majority of the false **compound splits** are cases of full **undersplitting**. The main reason for full **undersplitting** is a gold **lemma** which has no corpus evidence in the underlying training corpus. For example, the Afrikaans **compound** *kantoor|benodigdhede* ‘office requirement’ cannot be **split**, because the **head lemma** *benodigdhede* ‘requirement’ does not occur in Afrikaans WIKIPEDIA. The reason for false **compound splits** is also caused by data sparsity. If a correct **constituent** cannot be found in the training corpus, an alternative **compound split** (resulting from **word MOP**-based **normalization**) is produced.

**Conclusion** In this evaluation block, we compared the **compound splitting** performance for various external **compound splitters** on different gold standards. For the German **compound splitters**, we observed that ZvdP2016 is competitive with language-specific methods in the discipline of determining the correct **split points** (measured as *SPX*). In analogy to the results for the MOP set comparison, it turned out the ZvdP2016 is inferior to knowledge-rich methods in the discipline of **constituent normalization** due to misleading **word MOPs**. Another issue of ZvdP2016 is **undersplitting**: the recursive architecture of ZvdP2016 does not provide deeper **splitting** analyses having no top rank.

For Dutch and Afrikaans **compound splitting**, we presented **split point** accuracy numbers compared to the performance numbers published in Verhoeven et al. (2014). While ZvdP2016 outperforms VZDH2014 significantly for *Dutch*, our **word MOP**-based **splitter** is inferior for *Afrikaans*. In an error analysis, it turned out that the main reason for errors (i.e., **undersplitting** and **false splitting**) is data sparsity of the small Afrikaans training corpus (WIKIPEDIA). Thus, we can conclude that the **word MOP**-based approach is suitable for various languages but requires enough training data for learning both **constituent lemmas** and potential **constituent inflection** operations.

# 19. Semantically Informed Compound Splitting using Shallow Semantics

In this chapter, We present and elaborate parts of the work published in Ziering et al. (2016).

Most corpus-based [compound splitters](#) mainly rely on corpus frequency as key feature for estimating the correctness of a [compound split](#) (as discussed in Section 16.1). The [splitting](#) analysis having the [constituents](#) with the highest corpus frequency is usually ranked highest. As described in Section 15.1.1, approaches which are only based on corpus frequency are limited, because semantic plausibility is disregarded, i.e., the semantic compatibility between a [compound](#) and its [constituents](#), e.g., while the [compound](#) *Alleinstellung* ‘uniqueness’ can have the two possible analyses *Allein + Stellung* (lit: ‘alone + position’) and *All + Einstellung* ‘universe + attitude/adjustment’, the first [split](#) option is related to the **intended meaning** of the [compound](#).

In this section, We propose a flexible approach to enriching any [compound splitting](#) method that relies on corpus frequency with semantic information. In this approach, we measure the semantic similarity between the intended meaning of a [compound](#) and its [constituents](#), and promote [compound splits](#) for which this semantic similarity is highest. For example, while the [compound](#) *Eidotter* has the intended meaning of ‘egg yolk’ (related to the [split](#) *Ei | dotter*), an alternative [split](#) into valid [constituents](#) is *Eid | otter* with the meaning of ‘oath otter’. While a [splitting](#) approach relying on corpus-frequency as key feature would predict *Eid | otter* to be the correct [split](#) (due to the low frequency of *Dotter*), semantic information could reveal that there is a high semantic similarity between the intended meaning of *Ei* or *Dotter* and that of *Eidotter*. s

## 19.1. Distributional Semantics

### 19.1.1. Introduction

Below, we present an introduction into **Distributional Semantics (DS)**, a statistical semantic framework for representing lexical semantics and computing **word** similarity in terms of their contextual distribution. This introduction is partially borrowed from Ó Séaghdha (2008, p. 59). The abstract design of the following description keeps the **Distributional Semantics Model (DSM)** maximally general.

The basis of **DS** is the **distributional hypothesis** (Harris, 1954), saying that words occurring in the similar contexts tend to stand for a similar sense. Firth (1957) is known for the famous quotation:

*You shall know a word by the company it keeps*

In a **DSM**, the **Distributional Similarity (Dsim)** between a target **term**  $t_A$  and a target **term**  $t_B$  is estimated using their distributions in a certain context. For example, *beer* is distributionally more similar to *wine* than to *steak*, because they share contexts in which any beverage can occur (e.g., *a glass of <X>*, *drinking <X>*, ...), but in turn, *beer* is distributionally more similar to *steak* than to *computer*, because they share contexts in which any food **term** can occur (e.g., *he had <X> at dinner time*).

### 19.1.2. Formal Description

The target vocabulary  $\mathbf{V}_t$  comprises both target **terms**,  $t_A$  and  $t_B$ . The **Dsim** is based on a co-occurrence type  $c \in C$ , which is a pair of a co-occurrence relation  $r \in R$  and a co-occurrent **term**  $t_C$  out of a co-occurrent vocabulary  $\mathbf{V}_c$ , i.e.,  $C \subseteq R \times V_c$ .

The set of co-occurrence relations  $R$  comprises relations such as the unordered co-occurrence within a window of  $n$  **words** (i.e., *bag-of-words*), a syntactic function (e.g., *subject-verb*), **cross-lingual** alignments (e.g., translations), etc. The choice of co-occurrence relation  $r_i$  can yield different types of similarity: from the tight relatedness of synonymy (e.g., *beautiful*  $\sim$  *pretty*) to a looser relatedness of taxonomic similarity (e.g., co-hyponymy: *dog*  $\sim$  *cat*) or just a notion of general relatedness (Ó Séaghdha, 2008, p. 59), e.g., *computer*  $\sim$  *desktop*.

Target **terms** are represented as a vector of  $|C|$  dimensions, where each dimension corresponds to a co-occurrence type  $c \in C$  (i.e., a pair of co-occurrent **term**  $t_C$  and a co-occurrence relation  $r_i$  - in an **DSM**, different co-occurrence relations can be combined).

The value of each dimension is based on a weighting function  $g$  which maps the target term  $t_A$  and the co-occurrence type  $c$  on a real value, i.e.,  $g : V_t \times C \rightarrow \mathbb{R}$ . A possible value for  $g(t_A, c)$  is the co-occurrence frequency  $f(t_A \cap c)$ .

The similarity function  $Dsim^1$  maps two target terms  $(t_A, t_B)$  on a real value, i.e.,  $Dsim : \mathbb{R}^{|C|} \times \mathbb{R}^{|C|} \rightarrow \mathbb{R}$ . The commonly used metric for  $Dsim$  between the target terms  $t_A$  and  $t_B$  is the cosine similarity as shown in Formula 19.1.

$$Dsim(t_A, t_B) = cosine(\vec{t}_A, \vec{t}_B) = \frac{\vec{t}_A \cdot \vec{t}_B}{|\vec{t}_A| \cdot |\vec{t}_B|} = \frac{\sum_{k=1}^{|C|} t_{Ak} \cdot t_{Bk}}{\sqrt{\sum_{i=1}^{|C|} t_{Ai}^2} \cdot \sqrt{\sum_{i=1}^{|C|} t_{Bi}^2}} \quad (19.1)$$

## 19.2. Distributional Semantics for Compound Splitting

For measuring the semantic similarity between the intended meaning of a compound and that of its constituents, we use the  $Dsim$  between the distributional vector of compound and distributional vector of its potential constituents (i.e., modifier and head). For the example of the German compound *Eidotter* ‘egg yolk’, the assumption is that  $Dsim(\vec{Eidotter}, \vec{Ei}) > Dsim(\vec{Eidotter}, \vec{Eid})$  and that  $Dsim(\vec{Eidotter}, \vec{Dotter}) > Dsim(\vec{Eidotter}, \vec{Otter})$ .

This approach to measure the  $Dsim$  between a compound and its constituents has been proven to be a predictive metric for measuring the compositionality of a compound (Schulte im Walde et al., 2013, Weller et al., 2014). For example, while a compositional compound such as *Apfelsaft* ‘apple juice’ has a high  $Dsim$  to both constituents (*Apfel* ‘apple’ and *Saft* ‘juice’), a non-compositional compound such as *Maulwurf* ‘mole’ has a very low  $Dsim$  to alleged constituents (i.e., *Maul* ‘mouth’ and *Wurf* ‘toss’).

Our procedure of using  $Dsim$  for improving compound splitting can be interpreted as inverting the compositionality measurement of previous work. We are assuming that a target compound  $\Psi$  is compositional (which is true for most compounds), i.e., its true constituents  $\psi_i$  and  $\psi_j$  have a high  $Dsim$  to  $\Psi$ . In contrast, a wrong split (yielding to false constituents) would be non-compositional with respect to the intended meaning of  $\Psi$ , and would thus lead to a low  $Dsim$ . For example, if *Eid* | *Otter* would be the correct split for *Eidotter*, the interpretation would be non-compositional, because *Eid* ‘oath’ and *Otter* ‘otter’ are not related to the intended meaning of *Eidotter* ‘egg yolk’.

<sup>1</sup>For the sake of simplicity, we use  $Dsim$  for referring to the distributional similarity and to the formal similarity function.

## 19.3. Re-ranking Method

### 19.3.1. Initial Split Ranking

The proposed method is applicable to any **compound splitter** that produces a ranked output of **split options**<sup>2</sup> with their corresponding ranking score. For the re-ranking, we consider only corpus **lemmas** and no corpus **word forms**, i.e., only the **constituent lemmas** in the **LSF** are relevant.

For example, the **target compound** *Fischerzeugnis* having the intended meaning of ‘fish product’ is processed by a **compound splitter** yielding the output as given in Table 19.1. The top-ranked **LSF** is the result from a falsely triggered **normalization** rule. The suffix *er*, which is valid for **constituent lemmas** such as *Kind* ‘child’, is invalid for the **constituent lemma** *Fisch* ‘fish’. Furthermore, an *er* suffixation on *Fisch* would lead to the nominal derivation *Fischer* ‘fisherman’, an interpretation presented in the third line of Table 19.1.

Ranking score	LSF	Correctness
14264	<i>Fisch</i> + <i>Zeugnis</i> ‘fish certificate’	✗
9390	<i>Fisch</i> + <i>Erzeugnis</i> ‘fish product’	✓
5387	<i>Fischer</i> + <i>Zeugnis</i> ‘fisherman certificate’	✗

Table 19.1.: Initial split ranking

### 19.3.2. Determination of the Distributional Similarities

For each **LSF** of a **target compound**’s **split options** (e.g., *Fisch* + *Erzeugnis* given *Fischerzeugnis*), the **cosine similarity** between the **target compound**’s vector and each candidate **constituent**’s vector (i.e., both **modifier** and **head**) is determined, as a standard measure used for computing the **Dsim** (cf. Formula 19.1). Table 19.2 shows the **Dsim** values for all candidate splits presented in Table 19.1.

These **Dsim** values are used to predict the degree of semantic similarity between the intended meaning of the **target compound** and that of its candidate **constituents**.

<sup>2</sup>Following Weller et al. (2014), we focus on true **compounds** and ignore **non-split options**, i.e., **atomic analyses**.

Compound $\Omega$	Constituent $\omega$	$Dsim(\Omega, \omega)$
<i>Fischerzeugnis</i>	<i>Fisch</i>	0.46
<i>Fischerzeugnis</i>	<i>Erzeugnis</i>	0.10
<i>Fischerzeugnis</i>	<i>Fischer</i>	0.03
<i>Fischerzeugnis</i>	<i>Zeugnis</i>	0.01

Table 19.2.: Distributional similarity between **compound** and **constituents**

### 19.3.3. Distributional Similarity Modes

Besides directly using the **Dsim** values between **target compound** and the individual **constituents**, i.e., **modifiers** (**MOD**) and **head** (**HEAD**), we present experiments that use several ways to combine **modifiers** and **head** to different **Similarity Modes (SiModes)**.

Although the experiments outlined below are based on **binary compound splits**, the following **SiMode** formulas are designed for analyzing an  $N$ -ary **compound**  $\Psi$  ( $N \geq 2$ ), i.e., with any number of **constituents**,  $\psi_1, \dots, \psi_N$ .

**Geometric mean** As proposed by Weller et al. (2014), a possible combination of the candidate **constituents'** **Dsim** values is the geometric mean (GEO), as shown in Formula 19.2.

$$\text{GEO}(\psi_1 + \dots + \psi_N) = \sqrt[N]{\prod_{i=1}^N Dsim(\vec{\Psi}, \vec{\psi}_i)} \quad (19.2)$$

For example, let  $Dsim(\overrightarrow{\text{Fischerzeugnis}}, \overrightarrow{\text{Fisch}})$  be 0.455 and  $Dsim(\overrightarrow{\text{Fischerzeugnis}}, \overrightarrow{\text{Erzeugnis}})$  be 0.10. The GEO score for the **lemmas** of the **LSF**  $\text{Fisch} + \text{Erzeugnis}$  is  $\sqrt{0.455 \cdot 0.10} \approx 0.22$ . Table 19.3 shows the GEO scores for **modifier** and **head** of each **LSF** presented in Table 19.1.

<b>LSF</b>	<b>GEOScore</b>
<i>Fisch + Erzeugnis</i>	$\sqrt{0.46 \cdot 0.10} \approx 0.22$
<i>Fisch + Zeugnis</i>	$\sqrt{0.46 \cdot 0.01} \approx 0.05$
<i>Fischer + Zeugnis</i>	$\sqrt{0.03 \cdot 0.01} \approx 0.01$

Table 19.3.: Geometric mean **Dsim** scores

Moreover, we used standard arithmetic operations (Mitchell and Lapata, 2010, Widows, 2008) and combined the vectors of **modifiers** and **head** by vector addition (**ADD**),

and multiplication (MULT) as shown to be beneficial in Schulte im Walde et al. (2013), described below.

**Vector addition** An alternative way for combining **modifiers** and **head** is the vector sum. The ADD score is the **Dsim** value between the **target compound** vector  $\vec{\Psi}$  and the vector sum of **modifiers** and **head** ( $\sum_{i=1}^N \vec{\psi}_i$ ), as shown in Formula 19.3.

$$\text{ADD}(\psi_1 + \dots + \psi_N) = \text{Dsim}(\vec{\Psi}, \sum_{i=1}^N \vec{\psi}_i) \quad (19.3)$$

**Vector multiplication** In analogy to the vector addition, an alternative combination of **modifiers** and **head** is the vector multiplication. The MULT score is the **Dsim** value between the **target compound** vector  $\vec{\Psi}$  and the vector product of **modifiers** and **head** ( $\prod_{i=1}^N \vec{\psi}_i$ ), as shown in Formula 19.4.

$$\text{MULT}(\psi_1 + \dots + \psi_N) = \text{Dsim}(\vec{\Psi}, \prod_{i=1}^N \vec{\psi}_i) \quad (19.4)$$

#### 19.3.4. Split Score Product and Re-ranking

Although the GEO score for the **LSF** ranking presented in Table 19.3 already positions the correct **split** at the top, the ranking by pure **Dsim** information lacks the **lemmas'** corpus frequency which is a crucial information for **splitting** many **compounds**, as will be shown in Section 19.4.8. Therefore, in the last step, the initial **split** ranking scores are multiplied with the **SiMode** scores and finally, all **LSFs** are re-ranked accordingly. Table 19.4 shows the result from re-ranking the output presented in Table 19.1 using the enrichment with GEO scores.

Ranking score	LSF	Correctness
9390 · 0.22 ≈ <b>2034</b>	<i>Fisch + Erzeugnis</i> 'fish product'	✓
14264 · 0.05 ≈ <b>709</b>	<i>Fisch + Zeugnis</i> 'fish certificate'	✗
5387 · 0.01 ≈ <b>70</b>	<i>Fischer + Zeugnis</i> 'fisherman certificate'	✗

Table 19.4.: Split re-ranking with GEO scores



### 19.3.5. Data Sparsity Treatment

If there are no common co-occurrence types  $c \in C$  for a **known target compound**  $\Psi$  and its **known constituents**  $\psi_i$ , the *Dsim* value is 0.

A reason for why it is not possible to compute the *Dsim* values is data sparsity of the individual components, i.e., there is no corpus evidence for either the **target compound**  $\Psi$  or for at least one **constituent lemma**  $\psi_i$  among the proposed *LSFs*. As a result, no distributional vector can be compiled. Due to the high productivity of **compounds** (Section 3.3), having no corpus evidence is more likely to happen for **compounds** than for **constituents**.

If a **target compound**  $\Psi$  lacks corpus evidence, the related **compound splitting** cannot be re-ranked. In this case, the original frequency-based **split** score ranking is retained as back-off.

If the **target compound**  $\Psi$  has corpus evidence, while a potential **constituent lemma**  $\psi_i$  does not, there is no co-occurrence of  $\Psi$  and  $\psi_i$ , and thus the  $Dsim(\Psi, \psi_i)$  is set to 0. This case might never happen, since most **compound splitting** methods only propose **compound splits** (*LSFs* respectively) for which all **constituent lemmas** have corpus evidence<sup>3</sup>.

### 19.3.6. Non-compositional Compounds

Although we assume that **target compounds** (in particular those that require semantic support for **splitting**) are usually compositional (as discussed in Section 19.2) and although most downstream **NLP** methods do not require the **splitting** of non-compositional **compounds**, our re-ranker is also designed for **splitting** non-compositional **compounds**, e.g., for linguistic research on the compositionality of **compounds** (Schulte im Walde et al., 2013, Weller et al., 2014): fully non-compositional **compounds**, that do not require semantic support for **splitting** but still have **splitting** ambiguity, no **split** option would provide a significant semantic similarity between the intended meaning of **target compound** and that of its **constituents**. Thus, the system would fall back to a frequency-based **split** score, as described for data sparsity in Section 19.3.5.

---

<sup>3</sup>In the subsequent experiments, **compound splitting** and **split** re-ranking is performed on the same training corpus.

## 19.4. Experiments

### 19.4.1. Languages

In analogy to the [word MOP](#)-based [compound splitter](#) presented in Chapter 18, the proposed [LSF](#) re-ranking method is designed **language-independently**, i.e., the re-ranker can be applied to [compound splitting](#) methods designed for any [closed compounding language](#) (including the non-Germanic languages, for which approximating [constituent inflection](#) using [word inflection](#) is less reliable).

On the other hand, the quality of a [DSM](#) is sensitive to corpus size. The smaller the underlying training corpus, the smaller the impact of [split](#) re-ranking using [DS](#), because there are more [target compounds](#) without corpus evidence, as discussed in Section 19.3.5. Since the corpora provided for *Dutch* and *Afrikaans* (outlined in Section 18.6.2) are too small for illustrating the performance gain of the re-ranking method, we decided to present only results for *German*.

### 19.4.2. Training Corpus

For building the German [DSM](#), we used the same corpus and preprocessors as presented in Section 18.6.2, the German WIKIPEDIA and TREETAGGER. While we are aware of the fact that there are German corpora larger than WIKIPEDIA, which can increase the quality of the [DSM](#) (with respect to more representative distributional vectors and a higher coverage of the [target compounds](#) and the related [constituents](#)), we decided to apply the same corpus as presented in Section 18.6.2. By controlling for corpus size, it is possible to contrast the differences in [splitting](#) performance with respect to information type (i.e., [Dsim](#) vs. corpus frequency) irrespective of corpus size.

### 19.4.3. Evaluation Measurement

While we are working with a [split](#) ranking, the crucial item to be evaluated is the [split](#) option at **top position**. We presented several [intrinsic evaluation](#) metrics for [compound splitting](#) in Section 18.6.5. Since the re-ranker is intended to demote and promote true [compound splits](#), the evaluation category WN (i.e., unsplit [compounds](#)), is not relevant and all [atomic](#) analyses are discarded. Thus, precision, recall, accuracy and  $F_1$ -Score are identical. As discussed metric, we use **accuracy** for determining the correct [split points](#) (*SPAcc*) and **normalizing** the resulting [constituent forms](#) (*NormAcc*). For testing the **statistical significance** of the difference in performance between the initial [split](#)

ranking (INIT) and a re-ranking result, we use the **Approximate Randomization Test** (Yeh, 2000), with a significance level of  $p < 0.05$ .

#### 19.4.4. Gold Standard

In Section 18.6.4, we presented several **compound splitting** gold standards. Since the proposed re-ranking method is focusing on **constituent lemmas**, we decided to use only the gold standard developed by Henrich and Hinrichs (2011), HH2011GS, which contains only **lemmatized binary compounds**. While the experiments conducted in Section 18.6 include **target compounds** with **split point markers** (e.g., **hyphenated compounds**), these are excluded for evaluating the re-ranker yielding a data set of **52,937 target compounds**.

#### 19.4.5. Utilized Distributional Semantics Model

In analogy to the **DSM** of Weller et al. (2014), we adopted a setting whose parameters are tuned on a development set and proved best for the automatic rating of **compound compositionality** (Schulte im Walde et al., 2013). It employs corpus-based co-occurrence information extracted from a window of 20 **words** to the left and 20 to the right of a target **word**. We restricted to the 20,000 most frequent nominal co-occurents.

#### 19.4.6. Rankings in Comparison

We compared the performance of the initial ranking (INIT) of a **compound splitter**, based on all individual features (in particular, the corpus frequency of the proposed **constituents**), with the **splitting** performance after re-ranking by multiplying the initial ranking score with the selected **SiMode** score ( $RR_{\text{FREQ-DS}}$ ). The baseline ( $RR_{\text{DS}}$ ) is inspired by the aggressive **splitting** mode (DIST) of Weller et al. (2014): re-ranking of the unordered list of **LSFs** proposed by a **splitter** exclusively according to the **SiMode** score, i.e., the initial **split** score (including corpus frequency information) is disregarded. Finally, we show results for the re-ranking upper bound (UPPER): all analyses which the underlying **compound splitter** proposes are ranked at top position.

#### 19.4.7. Inspected Compound Splitters

We inspected the three German **compound splitters**, which have already been included in the experiment presented in Section 18.6: the **word MOP**-based approach presented in Chapter 18, ZvdP2016, the **SMOR**-based approach of Fritzinger and Fraser (2010), FF2010

and the updated version of the method developed by Weller and Heid (2012), WH2012, which uses a list of PoS-tagged **lemmas**, an extensive hand-crafted set of **normalization** rules and several hand-crafted corpus filters.

**System coverage** The re-ranking method is intended to be applied to a ranking output comprising two or more true (i.e., non-atomic) **binary split** options. Not all **compound splitters** provide several true **binary splitting** options for all **target compounds** in HH2011GS. Focusing on the impact of re-ranking, for each **compound splitter**, we considered only the **target compounds** for which the **splitters** produce at least two true **binary compound splits**. Since there is only one **binary split tree** for ZvdP2016, we considered the **binary splitting** decisions of the **target compound** (prior to recursion). For FF2010 and WH2012, we considered all **binary** analyses occurring in the system output ranking.

The gold standard subsets for all **splitters** are individual - since we did not aim to compare the systems against each other (which has already been done in Section 18.6), we did not evaluate on a common test set.

Table 19.5 shows the sizes of the three different test sets (second column). The knowledge-rich approach of FF2010 has the smallest test set, because many impossible analyses (e.g., due to falsely triggered **MOPs**) are already excluded. The most frequent type of **compounds** included in the test set of FF2010 are **three-Noun Compound (3NC)** such as *Haupt / anwendungs / gebiet* ‘main field of application’, having different **bracketings** (e.g., LEFT- or RIGHT-branched). Finally, we discarded unknown **compounds** for the  $RR_{DS}$  baseline. While the  $RR_{FREQ\cdot DS}$  re-ranker falls back on the initial frequency-based scores (as discussed in Section 19.3.5), for  $RR_{DS}$ , there is no information for re-ranking. For comparing  $RR_{DS}$  with  $RR_{FREQ\cdot DS}$  and with INIT on a common test set, we additionally DISCarded UnKnown **compounds** (**discUK**) for  $RR_{FREQ\cdot DS}$  and INIT. The third column in Table 19.5 shows the test set sizes after discarding unknown **compounds**.

System	Test set size	
	all	discUK
FF2010	5111	4121
WH2012	42,172	38,443
ZvdP2016	52,443	47,951
TOTAL	52,937	48,008

Table 19.5.: Test set coverage for **compound split** re-ranking

### 19.4.8. Results and Discussion

Tables 19.6, 19.7 and 19.8 show the results of **compound split** re-ranking applied to the three inspect **compound splitters** FF2010, WH2012 and ZvdP2016 for all **SiModes**.

#### General Trends

As a first result, for all inspected **compound splitters**, re-ranking by combining the initial **split** score with **DS** information ( $RR_{\text{FREQ-DS}}$ ) improves the initial **split** ranking (**INIT**) for at least one **SiMode** (i.e., **GEO**).

The **baseline** of using pure **SiMode** scores ( $RR_{\text{DS}}$ ) significantly (🔴) worsens the initial performance (**INIT**). This is in line with previous work (Weller et al., 2014) and shows that isolated semantic information does not suffice for **compound splitting** but needs to be introduced as an additional feature. In an error analysis, we observed that the corpus frequency, which is missing for  $RR_{\text{DS}}$ , is a crucial feature for **compound splitting** and helps to demote analyses based on typographical errors or unlikely **constituent normalization**. For example, while  $RR_{\text{FREQ-DS}}$  analyzes the **compound** *Haarwasser* ‘hair tonic’ with the correct and highly frequent **modifier** *Haar* ‘hair’,  $RR_{\text{DS}}$  selects the morphologically plausible but yet unlikely and infrequent verbal **modifier** *haaren* ‘to molt’, which happens to have the higher **Dsim** to *Haarwasser*.

Another kind of **splitting targets** that benefits from corpus frequency is the **3NC**. Binary **splitting** of **closed 3NCs** is comparable to the **parsing** of **open 3NCs** (which will be addressed in Part **E**). For example, the **compound** *Blind|darm|operation* ‘appendix operation’ (lit: ‘blind intestine operation’) is frequency-based correctly **split** into the **immediate constituents** *Blinddarm | operation* ‘[appendix] operation’, whereas  $RR_{\text{DS}}$  prefers the **RIGHT-branched bracketing** into *Blind | darmoperation* ‘blind [intestine operation]’. Since the rightmost **constituent** *Operation* ‘surgery/operation’ is more ambiguous, it has a smaller **Dsim** to the entire **compound** than the **RIGHT-branched complex constituent** *Darmoperation* ‘intestinal operation’. In contrast, the high corpus frequency of the non-compositional *Blinddarm* ‘appendix’ and of the **head** *Operation*, makes a frequency-based **compound splitter** choose the correct structure. On the other hand, the task of selecting the correct **bracketing** of a **3NC** can also benefit from semantic support. For example, using re-ranking by  $RR_{\text{FREQ-DS}}$ , the wrong **compound split** *Arbeits|platzmangel* ‘labor [lack of space]’ is corrected to *Arbeitsplatz|mangel* ‘job scarcity’. Therefore, we can conclude that the combination of corpus frequency and semantic plausibility (in terms of **Dsim**) is working best for **compound splitting**.

**Comparing the accuracy types**, we see that the determination of the correct **split point**, the easier discipline, achieves a  $SPAcc$  of 98.1% (GEO@INIT for WH2012, Table 19.7). However, there is only a small benefit for  $SPAcc$  when adding semantic support (e.g., +0.3% for WH2012). In contrast, **constituent normalization** (measured as  $NormAcc$ ) can be improved by +1.6% (GEO@RR<sub>FREQ-DS</sub>@discUK for ZvdP2016, Table 19.8).

**Comparing the SiModes**, we see that for **constituent normalization**, the more demanding discipline, that leads to the largest differences in performance (measured by  $NormAcc$ ) between the different **SiModes**, MOD outperforms HEAD (for RR<sub>FREQ-DS</sub>). For  $SPAcc$ , the trend is vice versa, i.e., the **SiMode** HEAD slightly outperforms MOD. Moreover, the **SiMode** GEO outperforms those based on **heads** or **modifiers** in isolation. A possible reason for this is that there are **target compounds** being semantically more similar to the **modifier** than to the **head** (e.g., *Baumschule* ‘tree farm’ (lit: ‘tree school’)) or vice versa (e.g., *Fliegenpilz* ‘toadstool’ (lit: ‘fly mushroom’)). Combining the similarity scores for both **constituent types** (e.g., using the geometric mean) caters for both types of **compounds**. In addition, we find that for  $NormAcc$ , the **SiMode** GEO outperforms the **SiModes** based on vector arithmetic (i.e., ADD and MULT, performing similarly), whereas for  $SPAcc$ , the performance between GEO and ADD/MULT is comparable.

### Individual Observations

**FF2010.** Although, we do not aim to compare the inspected **compound splitters** in this chapter, it is striking that FF2010 (Table 19.6), which is the most precise **splitting** method in Chapter 18 for HH2011GS, shows the poorest initial performance (INIT) on its test subset (about 10% worse in  $SPAcc$  than the other **splitters**), while the performance numbers for WH2012 and ZvdP2016 (Tables 19.7 and 19.8) are comparable.

The morphological analyzer SMOR, which provides only morphologically plausible **splitting** options (leading to an almost optimal UPPER bound), just leaves hard cases of **splitting** ambiguity for which a corpus frequency ranking approach (Koehn and Knight, 2003) is not sufficient. Actually, most of the wrong splits produced by FF2010 are also falsely analyzed by WH2012 and ZvdP2016. For example, the **bracketing** of 3NCs such as *Blei|kristall|glas* ‘lead crystal glass’, where the RIGHT-branching structure (i.e., *Blei + Kristallglas* ‘lead + crystal glass’) has a higher frequency score than the LEFT-branched structure required by the gold standard. In some cases, the gold standard **bracketing** can also be considered as debatable or both a LEFT- and RIGHT-branching structure is plausible (i.e., **semantic indeterminacy**, as discussed in Section 3.8.3). For example, the **compound** *Zinn|guss|erzeugnis* ‘tin casting product’ is **split** into the **immediate con-**

19. Semantically Informed Compound Splitting using Shallow Semantics

stituents *Zinn* ‘tin casting’ and *Erzeugnis* ‘product’, while the gold standard requires the immediate constituents *Zinn* ‘tin’ and *Gusserzeugnis* ‘cast product’.

In contrast, the knowledge-lean methods WH2012 and ZvdP2016 also produce morphologically implausible analyses, which can be disambiguated more easily using corpus frequency.

Metric	SP Acc					Norm Acc					
	MOD	HEAD	GEO	MULT	ADD	MOD	HEAD	GEO	MULT	ADD	
<b>SiMode</b>											
FF2010											
LITE	INIT	88.0				80.3					
	RR <sub>FREQ-DS</sub>	87.3	86.7	<b>88.7</b>	87.9	87.6	79.4	78.7	<b>80.9</b>	79.4	78.2
	UPPER	99.4				95.7					
MODCSP	INIT	88.4				79.8					
	RR <sub>DS</sub>	82.5	81.9	88.0	86.7	85.6	72.9	72.7	77.7	75.6	72.7
	RR <sub>FREQ-DS</sub>	87.6	86.8	<b>89.3</b>	88.3	87.9	78.6	77.8	<b>80.5</b>	78.7	77.2
	UPPER	99.3				95.6					

Table 19.6.: Results of *split* re-ranking for FF2010

When re-ranking (RR<sub>FREQ-DS</sub>) using information about *Dsim* between *compound* and *modifier* or *head* (MOD or HEAD), the performance of FF2010 significantly (♣) drops. In particular, the *SiMode* HEAD clearly underperforms (e.g., SP Acc drops about 1.3%). This is expected when considering the *bracketing* of 3NCs. As will be discussed in Part E, the majority *bracketing* class for a 3NC, A B C, is LEFT (i.e., [A B] C, the *LEFT class baseline*). However, there are many (LEFT-branched) 3NCs for which the complex *head* (i.e., B C) is more distributionally similar to A B C than only C is. One example for this is the *compound* *Blinddarmoperation* described above. Another example is given for the 3NC *Block|flöten|spieler* ‘recorder player’. Here, the complex *head* *Flötenspieler* ‘flute player’ is more distributionally similar to the *compound* than the simplex *head* *Spieler* ‘player’, while the complex *modifier* *Blockflöte* ‘recorder’ is more similar to the *compound* than the simplex *modifier* *Block* ‘block’. As a consequence, with respect to the majority class LEFT, the *SiMode* HEAD can worsen the *splitting* performance (because it promotes a RIGHT-branched structure for a LEFT-branched 3NC). On the other hand, in some LEFT-branched cases, a simplex *modifier* is more similar to the *compound* than the complex *modifier*. For example, for the LEFT-branched *Energie|spar|lampe* ‘energy-saving bulb’, *Energie* is more similar to the *compound* than *energiesparen* ‘to save energy’, while the simplex *head* *Lampe* is more similar to the *compound* than the

complex *head Sparlampe*. As a consequence, the **SiMode** MOD can also worsen the initial **splitting** performance.

The geometric mean of the **Dsim** between **compound** and **modifier/head** (i.e., GEO) combines the positive impact of both **SiModes** MOD and HEAD, and thus can improve the initial **splitting** performance. Due to the smaller test set size for FF2010, this improvement is not statistically significant. As will be shown for WH2012 and ZvdP2016, re-ranking with the **SiMode** GEO significantly improves the initial **split** ranking (INIT) for larger test sets.

Metric	SPAcc					NormAcc					
	MOD	HEAD	GEO	MULT	ADD	MOD	HEAD	GEO	MULT	ADD	
<b>SiMode</b>											
WH2012											
TLE	INIT	98.1					91.1				
	RR <sub>FREQ-DS</sub>	98.2 <sup>👍</sup>	98.2 <sup>👍</sup>	98.4 <sup>👍</sup>	98.3 <sup>👍</sup>	98.4 <sup>👍</sup>	91.6 <sup>👍</sup>	91.2 <sup>👍</sup>	91.6 <sup>👍</sup>	91.3 <sup>👍</sup>	91.5 <sup>👍</sup>
	UPPER	99.9					97.8				
discJW	INIT	98.2					91.5				
	RR <sub>DS</sub>	96.4 <sup>👎</sup>	96.5 <sup>👎</sup>	97.7 <sup>👎</sup>	96.9 <sup>👎</sup>	95.5 <sup>👎</sup>	85.2 <sup>👎</sup>	89.9 <sup>👎</sup>	86.4 <sup>👎</sup>	80.0 <sup>👎</sup>	70.6 <sup>👎</sup>
	RR <sub>FREQ-DS</sub>	98.4 <sup>👍</sup>	98.4 <sup>👍</sup>	98.5 <sup>👍</sup>	98.5 <sup>👍</sup>	98.5 <sup>👍</sup>	92.0 <sup>👍</sup>	91.6 <sup>👍</sup>	92.1 <sup>👍</sup>	91.8 <sup>👍</sup>	91.9 <sup>👍</sup>
	UPPER	99.9					98.2				

Table 19.7.: Results of **split** re-ranking for WH2012

**WH2012.** Besides the determination of the correct **bracketing**, outlined above, WH2012 also benefits from semantic support for **2NCs**. Weller and Heid (2012) exploited a hand-crafted set of morphological rules for **normalization**. Even restricted to only valid operations of **constituent inflection**, some rules are falsely triggered and lead to wrong splits. For example, the **compound** *Zahnseide* ‘dental floss’ is falsely **split** into *Zahn(s)|eide* ‘tooth oaths’ by truncating the *s*-suffix, which is not a valid operation for the **constituent lemma** *Zahn*. Since *Seide* ‘silk’ has a higher **Dsim** to *Zahnseide* than *Eid* ‘oath’, re-ranking (RR<sub>FREQ-DS</sub>) promotes the correct analysis: *Zahn | Seide*. For the large test set of WH2012, re-ranking (RR<sub>FREQ-DS</sub>) significantly (👍) outperforms the initial **split** ranking (INIT).

**ZvdP2016.** The **word MOP**-based **compound splitter** presented in Chapter 18 learns **constituent inflection** from **word inflection**. As a result, there are false **compound splits** which result from a morphological operation that occurs in **word inflection** but not in **constituent inflection**. For example, the **word MOP** *e/a* (as in the pluralized past tense



19. Semantically Informed Compound Splitting using Shallow Semantics

verb form of *sehen* ‘to see’: *sahen*) allows for the **compound** *Denkansatz* ‘intellectual approach’ to be **split** into *Denkan|satz* with the **constituent lemmas** *denken* ‘to think’ and *Satz* ‘sentence’. Re-ranking with  $RR_{\text{FREQ-DS}}$  promotes the correct **compound split** (*Denk|ansatz*) with the same **modifier lemma** *denken* and the **head lemma** *Ansatz* ‘approach’. For  $RR_{\text{FREQ-DS}}$ , all **SiModes** yield a significant (👍) improvement over the initial **split** ranking (INIT).

Metric		SPAcc					NormAcc				
		MOD	HEAD	GEO	MULT	ADD	MOD	HEAD	GEO	MULT	ADD
ZvdP2016											
T <sub>re</sub>	INIT	97.9					88.2				
	$RR_{\text{FREQ-DS}}$	98.0👍	98.2👍	98.3👍	98.2👍	98.3👍	89.5👍	88.4👍	89.7👍	89.1👍	89.1👍
	UPPER	99.8					97.4				
M <sub>csip</sub>	INIT	98.1					88.5				
	$RR_{\text{DS}}$	94.5👎	94.9👎	96.5👎	94.9👎	92.1👎	77.4👎	81.3👎	75.1👎	65.7👎	52.3👎
	$RR_{\text{FREQ-DS}}$	98.2👍	98.4👍	98.5👍	98.4👍	98.5👍	89.9👍	88.7👍	90.1👍	89.5👍	89.5👍
	UPPER	99.9					97.5				

Table 19.8.: Results of **split** re-ranking for ZvdP2016



# 20. Extrinsic Evaluation of Compound Splitting using Recognizing Textual Entailment

In this chapter, we present and elaborate parts of the work published in Jagfeld et al. (2017).

We propose a novel way for [extrinsically](#) evaluating [compound splitting](#). While [Statistical Machine Translation \(SMT\)](#) is commonly used as [extrinsic evaluation](#) method, we argue for using the task of [Recognizing Textual Entailment \(RTE\)](#), a method which has several advantages over [SMT](#) (to be discussed in [Section 20.3](#)).

## 20.1. Introduction

### 20.1.1. Textual Entailment

The relation of [Textual Entailment \(TE\)](#) is a directional relationship between an *entailing* text fragment  $T$  and an *entailed* hypothesis,  $H$ , saying that the meaning of  $T$  entails (or infers) the meaning of  $H$  denoted as  $T \Rightarrow H$ . This relation holds if “typically, a human, reading  $T$ , would infer that  $H$  is most likely true” (Dagan et al., 2006). Entailment is directed, i.e.,  $T \Rightarrow H$  does not mean that  $H \Rightarrow T$ . For example, while the verb *buy* entails *own*, the entailment in the opposite direction is unlikely (Dagan and Glickman, 2004). A non-entailment, i.e., the fact that  $T$  does not entail  $H$ , is denoted as  $T \not\Rightarrow H$ . A third type of entailment, in which  $T$  contradicts  $H$ , can be reduced to the positive entailment relation between  $T$  and the negation of  $H$ ,  $T \Rightarrow \neg H$ . [Table 20.1](#) shows some examples of entailment and non-entailment pairs.

[Recognizing Textual Entailment \(RTE\)](#) is a [binary](#) classification on the decision whether a given text  $T$  entails a given hypothesis  $H$ .

There is a strong variation in the correlation between linguistic expressions and the

Text $T$	Hypothesis $H$	Relation
<i>Peter <b>studies</b> NLP</i>	<i>Peter is a <b>student</b></i>	$T \Rightarrow H$ (ENTAILMENT)
<i>Peter buys a computer</i>	<i>Peter has a wife</i>	$T \not\Rightarrow H$ (NON-ENTAILMENT)
<i>Mary's <b>husband</b> sleeps</i>	<i>Mary is <b>single</b></i>	$T \Rightarrow \neg H$ (CONTRADICTION)

Table 20.1.: Examples for [Textual Entailment \(TE\)](#)

underlying sense. There are different ways of expressing a certain meaning (e.g., by using synonyms on the lexical level), and in contrast, there are different meanings for one and the same linguistic expressions (e.g., due to [word](#) sense ambiguity). These ‘*meaning-preserving linguistic variations*’ (Zeller, 2016) have to be distinguished from  $T$ - $H$  pairs denoting a difference in meaning by an [RTE](#) system.

### 20.1.2. The Benefits of RTE for various NLP Tasks

[RTE](#) is relevant for many [NLP](#) tasks such as [Information Extraction \(IE\)](#), [Information Retrieval \(IR\)](#), [Machine Translation \(MT\)](#), multi-document summarization or [Question Answering \(QA\)](#). As an example from a [QA](#) system, Dagan et al. (2009) discuss the following example: for the question *Who is John Lennon’s widow?*, a [QA](#) system has to know that the text *Yoko Ono unveiled a bronze statue of her late husband, John Lennon, . . .* infers the answer *Yoko Ono is John Lennon’s widow*. In an [IR](#) system, the semantic concepts denoted by a given query have to be entailed from relevant documents to be retrieved (Dagan and Glickman, 2004).

### 20.1.3. The Lexical Overlap Approach

The [RTE](#) approach presented in this thesis aims to map as much lexical material of  $H$  to  $T$  as possible. It is based on an assumption presented in Zeller (2016, p. 157): the higher the coverage of lexical material of  $H$  in  $T$  (subsequently denoted as [H coverage](#)), the more likely  $T \Rightarrow H$ , rather than  $T \not\Rightarrow H$ . In the following sections, we will interpret ‘lexical material of  $H$ ’ as the set of [atomic lexemes](#) used for the proposition of  $H$ . All subsequent mentions of [RTE](#) include the lexical overlap approach.

### 20.1.4. Outline of this Chapter

In Section 20.2, we discuss some limitations of [RTE](#) systems (20.2.1), how these can be overcome using [compound splitting](#) in advance (20.2.2) and vice versa how [RTE](#) can

be used for evaluating [compound splitting extrinsically](#) (20.2.3). In Section 20.3, we compare the pros and cons of [extrinsically evaluating compound splitting](#) using SMT and RTE. Section 20.4 outlines a language-independent multi-level alignment framework for RTE and an entailment algorithm, proposed by Noh et al. (2015), which will be the basis for the experiments and results for evaluating various [compound splitters](#) on RTE, as presented in Section 20.5.

## 20.2. RTE and Compound Splitting

In this section, we describe the symbiotic correlation between RTE and [compound splitting](#), i.e., in how far RTE can be used for [compound splitting](#) and vice versa how [compound splitting](#) can be beneficial for RTE.

### 20.2.1. Limitation due to Opacity of Closed Compounds

Switching to German RTE, a crucial limitation is the opacity of [closed compounds](#): there is no information about the composed [constituents](#). One consequence is that [closed compounds](#) and their [constituents](#) cannot be matched as an indicator for TE (based on the lexical overlap approach, described in Section 20.1.3), as shown in the examples below.

- (6) a. *T*: *Peter kauft ein Kinder**buch*** ‘Peter buys a children’s **book**’  
 b. *H*: *Der Händler verkauft Peter ein **Buch*** ‘The retailer sells Peter a **book**’
- (7) a. *T*: *Der Pilot überfliegt **Berge*** ‘The pilot crosses **mountains**’  
 b. *H*: *Der Jet passiert eine **Bergkette*** ‘The jet passes a **mountain** chain’
- (8) a. *T*: *Kinder lieben **Fruchtsäfte** aus **Äpfeln***  
 ‘Children love fruit **juices** made of **apples**’  
 b. *H*: *Peter’s Sohn liebt **Apfelsaft*** ‘Peter’s son loves **apple juice**’

Example 6 shows an entailing *T-H* pair for which no correlation between the compound *Kinderbuch* ‘children’s book’ and its [head](#) *Buch* ‘book’ can be found as entailment indicator. Example 7 illustrates that this problem also occurs for [compound modifiers](#), e.g., the [compound](#) *Bergkette* ‘mountain chain’ cannot be matched with *Berge* ‘mountains’. Finally, Example 8 combines the first two problems: here, the [atomic word](#) *Äpfeln* ‘apples’ in *T* cannot be matched with the [modifier](#) of the [compound](#) *Apfelsaft*

‘apple juice’ in  $H$ , and the **head** of the **compound** *Fruchtsäfte* ‘fruit juices’ in  $T$  cannot be matched with the **head** of the **compound** *Apfelsaft* in  $H$ .

Moreover, the opacity of **closed compounds** hides the true number of uncovered **lexemes** in  $H$ . Example 9 shows a non-entailing  $T$ - $H$  pair in which there are two covered  $H$  tokens (*Peter* and *liest*) without prior **compound splitting**, leading to an **H coverage** of  $\frac{2}{4} = 0.5$ . After splitting the **compound** *Bücher|regal|aufbau|anleitung* ‘bookshelf assembly instructions’ in  $H$ , the number of covered **tokens** increases by one, but the total number of **tokens** in  $H$  increases by 3, leading to an **H coverage** of  $\frac{2+1}{4+3} = \frac{3}{7} \approx 0.43$ , promoting the correct  $T \not\equiv H$  classification.

- (9) a.  $T$ : *Peter liest ein **Buch*** ‘Peter reads a **book**’  
 b.  $H$ : *Peter liest die **Bücherregal****aufbau****anleitung***  
 ‘Peter reads the **bookshelf** assembly instructions’

### 20.2.2. Enriching RTE with Compound Splitting

For overcoming the limitation of **RTE** systems due to the opacity of **closed compounds**, we aim for enriching **RTE** with **compound splitting** information. Considering the lexical overlap approach, outlined in Section 20.1.3, the goal is to establish an alignment between **atomic terms** or the **constituents** of a **closed compound** in  $H$  and the **atomic terms** or **constituents** of a **closed compound** in  $T$ , and to reveal the number of **atomic lexemes** occurring in  $H$ , for having a justified increase or decrease of the **H coverage**.

A possible approach to enriching **RTE** with **compound splitting** is to apply all **terms** in  $T$  and  $H$  to a **compound splitter** prior to **RTE**, i.e., to replace a potential **closed compound** with all of its potential **constituents** (i.e., with an **open compound** variant or a **lemma sequence format** (LSF)).

As shown in the examples 6, 7 and 8, there is need to provide both **modifiers** and **head** in the **open compound** variant. First experiments showed that replacing a **closed compound** with only the **head** performs worse. The **RTE** framework, which will be discussed in Section 20.4, includes a **lemmatization** step for preprocessing the (**split**) input data. Since the utilized **lemmatizer** is trained on **word inflection** (rather than **constituent inflection**), the replacements in  $T$  and  $H$  are **constituent lemmas** (as provided with the **LSF** output of a **compound splitter**). For example, the **closed compound** *Kindheitserinnerung* ‘childhood memory’ is replaced by the **open compound** variant (or **LSF**) *Kindheit Erinnerung*.

### 20.2.3. RTE as Extrinsic Evaluation Method

A good evaluation method for [compound splitting](#) has to penalize all errors a [compound splitter](#) can produce: [undersplitting](#), [false splitting](#) and [oversplitting](#). Moreover, correct [splitting](#) needs to be rewarded. Below, we describe how RTE based on the lexical overlap approach (as described in Section 20.1.3) can be used for the [extrinsic evaluation](#) of [compound splitting](#).

#### Impact of Correct Splitting

If a [closed compound](#) in  $H$  is [split](#) into its true [constituents](#), it contains all (composed) [lexemes](#) used for the proposition of  $H$ , and provides transparent [tokens](#) to be matched with [tokens](#) in  $T$ . For correct [compound splits](#) in  $T$ , the set of [tokens](#) in  $T$  expands, allowing for matches with [tokens](#) in  $H$ . As described in Section 20.2.1, correct [compound splitting](#) improves the quality of the RTE performance, with respect to both  $T \Rightarrow H$  and  $T \not\Rightarrow H$ .

There is a **limitation** of measuring correct [splitting](#). The number of uncovered [tokens](#) in  $H$  can also increase for valid [compound splitting](#) if the resulting [constituents](#) exclusively occur in  $H$ , as shown in Example 10 and Example 11.

- (10) a.  $T$ : Der Pilot fliegt in einem Jet ‘The pilot controls a jet’  
 b.  $H$ : Der {Flug Kapitän} fliegt ein {Flug Zeug}  
 ‘The aircraft captain controls an airplane’
- (11) a.  $T$ : Peter fährt einen Mercedes ‘Peter drives a Mercedes’  
 b.  $H$ : Peter ist ein {Auto Fahrer} ‘Peter is a car driver’

In Example 10, the [atomic word](#) *Pilot* is synonymous to the [compound](#) *Flugkapitän* and *Jet* is a hyponym of the [compound](#) *Flugzeug*. Integrating semantic knowledge prior to [compound splitting](#), the entailing relation in Example 10 could be revealed.

In Example 11, the [modifier](#) *Auto* in  $H$  is a hypernym of *Mercedes* in  $T$  and the [head](#) *Fahrer* in  $H$  is a derivation of *fährt* (or *fahren*) in  $T$ . Matching semantically and derivationally related [lexemes](#) between  $T$  and  $H$  after [compound splitting](#), the entailing relation between  $T$  and  $H$  can be determined.

While we expect a mitigation of these limitations, when adding further knowledge, in the experiments outlined below, we will not combine [compound splitting](#) with semantic or derivational information. We will investigate different ways of combining [compound splitting](#) and further lexical knowledge resources in future work.

### Impact of Undersplitting

In the case of **undersplitting**, **closed compounds** retain opaque. This means, the limitations due to **compound** opacity outlined above are not solved and the **RTE** performance remains equal to the performance of the initial **RTE** setup. Although this erroneous behavior of a **compound splitter** cannot be measured in isolation (because an **RTE** dataset does not necessarily include **closed compounds**), **undersplitting** becomes apparent when comparing various **compound splitters**, where one **splitter** achieves less improvement than other **splitters**.

### Impact of Oversplitting

In the case of **oversplitting lexemes** exclusively occurring in  $H$ , the number of uncovered **tokens** in  $H$  increases (i.e., the **H coverage** decreases). This means that **oversplitting** has a negative impact on the **RTE** performance, because for entailing  $T$ - $H$  pairs the  $T \not\Rightarrow H$  class is promoted.

However, there are some **limitations** for measuring **oversplitting**. When **oversplitting lexemes** occurring both in  $T$  and  $H$ , the **H coverage** increases, giving common **tokens** more weight. Moreover, **oversplitting lexemes** exclusively occurring in  $T$  does not have any impact on the **RTE** performance.

### Impact of False Splitting

The **false splitting** of **closed compounds** exclusively occurring in  $H$  leads to the same result like **oversplitting**: the **H coverage** decreases, because none of the potential **constituents** can be aligned to a **token** in  $T$ , while **constituents** from an alternative **compound split** of the **target compound** could have, as shown in Example 12. The falsely tagged hypothesis  $H_2$  has a lower **H coverage**, because there is no **constituent lemma** aligning to *Dottern* in  $T$ , whereas the correct **splitting** in  $H_1$  leads to a higher **H coverage**.

- (12) a.  $T$ : *Peter isst abends ein Omelett mit zwei Dottern*  
 ‘Peter has an omelet with **two yolks** for **dinner**’
- b.  $H_1$  ✓: *Peters {Abend essen} enthielt zwei Ei Dotter*  
 ‘Peter’s **dinner** includes **two** egg **yolks**’
- c.  $H_2$  ✗: *Peters Abend essen enthielt zwei Eid Otter*  
 ‘Peter’s **dinner** includes **two** oath otters’



While the correct **splitting** of *Abend|essen* ‘dinner’ and *Ei|dotter* ‘egg yolk’ in  $H_1$  leads to a **token** coverage of  $\frac{5}{7} \approx 71\%$  (assuming that *essen* ‘meal’ is aligned to the derived verb *isst* ‘eats’ and *abends* ‘in the evening’ is aligned to the noun *Abend* ‘evening’), the **false splitting** in  $H_2$  leads to the smaller coverage of  $\frac{4}{7} \approx 57\%$ .

### 20.3. Extrinsic Evaluation: SMT vs. RTE

Besides **intrinsic evaluation** methods (based on gold standards), previous work on **compound splitting** tests the performance **extrinsically** mainly on the task of **SMT** (as discussed in Chapter 16). This task can benefit from prior **compound splitting** when a **closed compound** (frequently unknown to a translation dictionary due to the productivity of **compounds** (see Section 3.3)) is to be translated, e.g., in a German-to-English translation. Translating the composed **constituents** in isolation often yields a correct **open compound** equivalent. For example, translating the (probably unknown) deictic **closed compound** *Gurkentisch* ‘cucumber table’ would fail unless it is **split** into the **constituents** *Gurke* ‘cucumber’ and *Tisch* ‘table’, beforehand. A **compound splitter** can be evaluated **extrinsically** by comparing the performance of a downstream **SMT** system with and without prior **splitting** of **compounds**.

As discussed in Section 20.2.3, **RTE** as an **extrinsic evaluation** task for **compound splitting** is a promising alternative for **SMT** which provides various advantages, some of which are discussed below.

#### Oversplitting

One source of errors of **compound splitters** can be **oversplitting**, i.e., **atomic lexemes** are mistakenly **split**. As observed by Dyer (2009), Fritzingler and Fraser (2010) and Weller et al. (2014), phrase-based **SMT** is robust to **oversplitting**, because oversplit **terms** (i.e., sequences of hypothesized **constituents**) can be learned as phrases. As has been discussed in Section 20.2.3, while our approach to integrating **compound splitting** to **RTE** has also some robustness issues with respect to **oversplitting**, it appears more suitable for **terms** occurring exclusively in  $H$ .

#### Gold Agreement

As discussed by Olive et al. (2011, sec. 5.1.2), the “*general notion of quality of a translation is [...] subjective*”. As for many **NLP** tasks including language generation, there

are almost infinitely many possible “correct” outputs, whereas there are only a finite number of reference translations given in an MT gold standard. In contrast, RTE is a clearly defined binary classification task and there is a high agreement in the entailment decisions. For the RTE test set used in the following experiments, there is an average agreement rate of 87.8% with an average  $\kappa$  level of 0.75 (Giampiccolo et al., 2007), meaning substantial agreement (Landis and Koch, 1977).

## Natural Language Understanding

To the best of our knowledge, we are the first to use RTE as extrinsic evaluation method for compound splitting, despite the fact that RTE is a promising alternative for compound splitting: both RTE and the compound splitting directly serve for the task of Natural Language Understanding (NLU).

## Transparency of RTE

Commonly used SMT methods (e.g., the MOSES toolkit (Koehn et al., 2007)) are computationally complex. The SMT result is a full translation of large amounts of text (whereas only the compound translation would be necessary for the evaluation). As a consequence, there is only a minor difference in BLEU score between different splitting approaches (Escartín, 2014). In the subsequent experiments, we will avoid the usage of neural RTE systems (Bowman et al., 2015) because of the opacity of the models, which will make interpretation of the effect harder. Instead, we will make use of a transparent RTE system (which will be described in Section 20.4) that allows for better estimating the impact of compound splitting on RTE, i.e., we can trace back which  $T$ - $H$  pair has been changed during splitting and whose  $T$ - $H$  pair’s entailment classification has changed because of which RTE feature.

## 20.4. Multi-level Alignment Framework

The basis of the proposed RTE system, utilized in this thesis, is a modular and extensible framework for RTE with alignments between  $T$  and  $H$  on several levels of analysis, presented by Noh et al. (2015). Figure 20.1 shows the dataflow of the multi-level alignment architecture proposed by Noh et al. (2015).

The framework includes four steps towards TE classification.

1. **Linguistic pre-processing** First, the text  $T$  and the hypothesis  $H$  are linguistically pre-processed (e.g., **tokenized**, **PoS-tagged**, **lemmatized**, ...).
2. **Alignment step** In the next step, several aligners supported by knowledge sources (e.g., hyponymy relations from WordNet (Miller, 1995b)) are applied to the  $T$ - $H$  pair. The alignments can be set between different levels of analysis (e.g., **atomic lexemes** or sequences). All alignments are combined into a *multi-level alignment* representation.
3. **Feature extraction** In the third step, features are extracted from all alignment levels and stored in feature vectors representing the  $T$ - $H$  pair.
4. **Entailment classification** Finally, the feature vectors are used in a supervised entailment classifier.

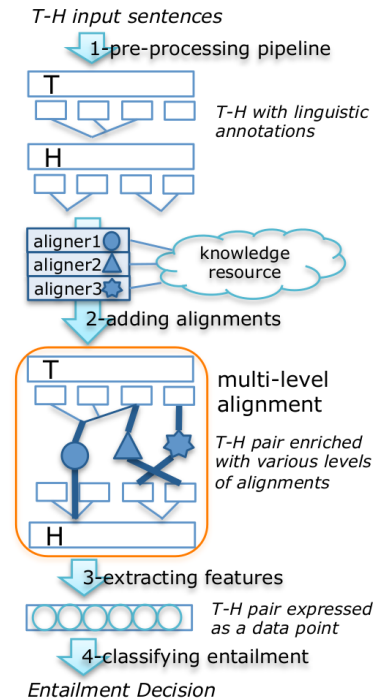


Figure 20.1.: Dataflow proposed by Noh et al. (2015)

On the basis of the multi-level alignment framework, Noh et al. (2015) present a publicly available<sup>1</sup> entailment decision algorithm (EDA). It is part of the Excitement Open Platform (EOP) for TE (Padó et al., 2015), which includes **multilingual** preprocessors (e.g., TreeTagger (Schmid, 1995)) and knowledge resources (e.g., English WordNet (Miller, 1995b) or German DErivBase (Zeller et al., 2013)).

Noh et al. (2015) propose three possible aligners:

1. **Lexical aligner:** A link between **tokens** in  $T$  and  $H$  is set if a given lexical resource (e.g., GermaNet (Hamp and Feldweg, 1997, Henrich and Hinrichs, 2010)) points to a relation between them.
2. **Paraphrase aligner:** This aligner sets a link between (lists of) **tokens** in  $T$  and  $H$  if these are related in a paraphrase resource (e.g., a paraphrase table).

<sup>1</sup><https://github.com/hltfbk/EOP-1.2.1/wiki/AlignmentEDAP1>

**3. Lemma identity aligner:** Based on the [lemmas](#) produced by the preprocessor, this aligner sets a link between [tokens](#) in  $T$  and  $H$  if they are tagged with the same [lemma](#).

Finally, Noh et al. (2015) propose four features that are measuring the [H coverage](#) with respect to various [token](#) types: (1) [H coverage](#) of any [tokens](#), (2) [H coverage](#) of [content words](#), (3) [H coverage](#) of verbs and (4) [H coverage](#) of named entities (given a [PoS](#) tag set which indicates named entities).

## 20.5. Experiment

### 20.5.1. Simplified Algorithm

For using [RTE](#) as [extrinsic evaluation](#) method for [compound splitting](#), a simple version of the [RTE](#) algorithm P1EDA, proposed by Noh et al. (2015), is used. The simplification of the algorithm is motivated by the fact that the goal of the [extrinsic evaluation](#) method is not to yield best [RTE](#) performance, but to be able to measure fine-grained differences between [RTE](#) with and without prior [compound splitting](#) produced by various [splitting](#) methods. Therefore, we decided to use only the [lemma](#) identity aligner based on TREE-TAGGER corpus [lemmas](#). While Noh et al. (2015) proposed to use verb coverage as feature, we excluded it from the German setup, because this feature is said to have no good performance for German verbs in the EOP P1EDA code documentation<sup>2</sup>.

### 20.5.2. Training and Test Set

In the recent years, there have been several benchmarking workshops on [RTE](#), the **PASCAL Recognising Textual Entailment Challenges** (Dagan et al., 2006). Each workshop has published an [RTE](#) dataset containing both training and test set of  $T$ - $H$  pairs, manually labelled with  $T \Rightarrow H$  and  $T \not\Rightarrow H$ . For the experiments described below, the training and test set derived from the third PASCAL challenge, RTE-3 (Giampiccolo et al., 2007) is used, which contains 800  $T$ - $H$  pairs for training and 800  $T$ - $H$  pairs for testing. In both sets, the classes  $T \Rightarrow H$  and  $T \not\Rightarrow H$  are balanced (leading to 50% accuracy as chance baseline). Initially, this [RTE](#) gold standard was only for *English*. Magnini et al. (2014) manually translated the RTE-3 dataset to *German* and *Italian*. For German [compound splitting](#), the German version of RTE-3 is used.

<sup>2</sup><https://github.com/hltfbk/Excitement-Open-Platform/blob/master/alignmentedas/src/main/java/eu/excitementproject/eop/alignmentedas/p1eda/instances/SimpleWordCoverageDE.java>

### 20.5.3. Supervised Classification

For P1EDA, a **multinomial logistic regression classifier** has been used (Magnolini and Magnini, 2015). It is trained on the training set, which is **split** using the various **compound splitters**, i.e., each **compound splitter** creates its own model. The trained RTE system is applied to the test set, which is also **split** using the corresponding **compound splitter**.

### 20.5.4. RTE Evaluation Measurements

For measuring the performance of the RTE system (with and without prior **compound splitting**), the standard measures for evaluating RTE (Formulas 20.1 to 20.7) are used for the classes  $T \Rightarrow H$  and  $T \not\Rightarrow H$ . These are based on the four categories

- (a) **T  $\Rightarrow$  H  $\checkmark$**  Number of  $T$ - $H$  pairs that are correctly classified as  $T \Rightarrow H$
- (b) **T  $\Rightarrow$  H  $\times$**  Number of  $T$ - $H$  pairs that are wrongly classified as  $T \Rightarrow H$
- (c) **T  $\not\Rightarrow$  H  $\checkmark$**  Number of  $T$ - $H$  pairs that are correctly classified as  $T \not\Rightarrow H$
- (d) **T  $\not\Rightarrow$  H  $\times$**  Number of  $T$ - $H$  pairs that are wrongly classified as  $T \not\Rightarrow H$

The accuracy ( $Acc$ ) is the ratio of the number of all correctly classified  $T$ - $H$  pairs divided by the number of all  $T$ - $H$  pairs.

$$Acc = \frac{T \Rightarrow H \checkmark + T \not\Rightarrow H \checkmark}{T \Rightarrow H \checkmark + T \not\Rightarrow H \checkmark + T \Rightarrow H \times + T \not\Rightarrow H \times} \quad (20.1)$$

The precision ( $P$ ), the recall ( $R$ ) and the resulting  $F_1$  score is computed for each class separately.

$$P_{T \Rightarrow H} = \frac{T \Rightarrow H \checkmark}{T \Rightarrow H \checkmark + T \Rightarrow H \times} \quad (20.2)$$

$$P_{T \not\Rightarrow H} = \frac{T \not\Rightarrow H \checkmark}{T \not\Rightarrow H \checkmark + T \not\Rightarrow H \times} \quad (20.3)$$

$$R_{T \Rightarrow H} = \frac{T \Rightarrow H \checkmark}{T \Rightarrow H \checkmark + T \not\Rightarrow H \times} \quad (20.4)$$

$$R_{T \not\Rightarrow H} = \frac{T \not\Rightarrow H \checkmark}{T \not\Rightarrow H \checkmark + T \Rightarrow H \times} \quad (20.5)$$

$$F_{1T \Rightarrow H} = \frac{2 \cdot P_{T \Rightarrow H} \cdot R_{T \Rightarrow H}}{P_{T \Rightarrow H} + R_{T \Rightarrow H}} \quad (20.6)$$

$$F_{1T \not\Rightarrow H} = \frac{2 \cdot P_{T \not\Rightarrow H} \cdot R_{T \not\Rightarrow H}}{P_{T \not\Rightarrow H} + R_{T \not\Rightarrow H}} \quad (20.7)$$

Since the accuracy of the two classes  $T \Rightarrow H$  and  $T \not\Rightarrow H$  as well as micro-averaged precision, recall and  $F_1$  are equal, we will only report results for one accuracy measure.

For testing statistical significance, we use McNemar’s test (McNemar, 1947) on a significance level of  $p < 0.05$ .

### 20.5.5. Target Languages

Although Noh et al. (2015) designed the multi-level alignment framework as language-independent and evaluated RTE on all three languages of the RTE-3 dataset (i.e., *English*, *German* and *Italian*), we restrict to *German* as the only closed compounding language, subject to compound splitting, among the RTE-3 languages. To the best of our knowledge, there are no RTE gold standards for the other closed compounding languages presented within this thesis (i.e., *Dutch* and *Afrikaans*). The Cross-Language Evaluation Forum (CLEF) provides Dutch data for the similar task of Answer Validation Exercise<sup>3</sup> (AVE), which uses the same RTE formalism for pushing QA (Bikel and Zitouni, 2012, sec. 6.2.5). We will investigate the usability of AVE data for RTE-based evaluation of compound splitting in future work.

### 20.5.6. Inspected Compound Splitters

For the extrinsic evaluation, all inspected compound splitters from Chapter 18 and Chapter 19 are used:

FF2010: The SMOR-based compound splitting approach developed by Fritzinger and Fraser (2010)

WH2012: The corpus-based approach including an extensive set of morphological transformation rules and a resource-based filtering of corpus lemmas, developed by Weller and Heid (2012)

ZvdP2016: The word MOP-based approach of Ziering and Van der Plas (2016), presented in Chapter 18

### 20.5.7. True Casing of Constituent Lemmas

While the splitting output of FF2010 is true-cased (e.g., nouns are capitalized, whereas verbs and adjectives are lower-cased), the (optionally) PoS-tagged output of the constituent lemmas in WH2012 and ZvdP2016 is lower-cased for all word categories.

---

<sup>3</sup><http://nlp.uned.es/clef-qa/ave/>

Since we want to have a consistent PoS sequence for  $T$  and  $H$ , we replace potential **closed compounds** with the corresponding **open compound** variants prior to the application of the EOP-internal TREE\_TAGGER, rather than using the PoS tags provided by the **compound splitters**. In order to get the correct named entity PoS tag (for catering the named entity **H coverage** feature), and to get the correct **lemmas** (e.g., *feuer* ‘to fire (imperative)’ is **lemmatized** to *feuern* ‘to fire (infinitive)’, whereas *Feuer* ‘fire’ is already the correct **lemma**), we capitalize all relevant predicted **constituent lemmas** (i.e., nouns and named entities) of WH2012 and ZvdP2016 (based on the PoS tags of the **constituent forms**) prior to the replacement in  $T$  and  $H$ . Since capitalized nouns are a German-specific property, we do not include the true-casing step in the language-independent **compound splitting** method of ZvdP2016, outlined in Chapter 18, but apply this transformation separately in the current RTE experiment focusing to *German*.

### 20.5.8. Adding Compounding vs. Derivational Information

Recently, Zeller et al. (2013) developed the high-coverage resource for German morphological derivation, DERivBase. It groups 280K **lemmas** into 17K *derivational families*, i.e., clusters of derivationally related **lemmas**. For example, the **lemma** *Sport* ‘sport’ is mapped onto the following family: {*Sportler* ‘sportsman’, *sportiv* ‘sporty’, *sportlich* ‘sporting’, *sportmäßig* ‘athletic’, *Sportlandschaft* ‘sports landscape’, *Sportlerin* ‘sportswoman’, *Sportlichkeit* ‘sportiness’, *unsportlich* ‘unsportsmanlike’, ...}. Zeller (2016) evaluated DERivBase by enriching an entailment algorithm with a DERivBase lexical aligner (see Section 20.4). This lexical aligner maps a **token**  $h_i$  in  $H$  onto a **token**  $t_i$  in  $T$  if  $h_i$  and  $t_i$  share a common derivational family.

In order to reveal the difference between adding derivational and **compounding** information to an RTE system, we compare the RTE setups enriched with **compound splitting** against an RTE setup in which an additional lexical aligner based on DERivBase, v1.4 (Zeller et al., 2013) is used.

## 20.6. Results

Table 20.2 shows the results for the initial RTE performance (INIT) and the RTE performance with prior **compound splitting**, executed by the three inspected **splitters** (ZvdP2016, FF2010 and WH2012), as well as for the addition of a DERivBase lexical aligner.

System	Acc	T⇒H			T≠H		
		P	R	F <sub>1</sub>	P	R	F <sub>1</sub>
INIT	64.13	62.50	74.57	68.00	66.67	53.20	59.17
ZvdP2016	66.63	64.55	77.02	70.23	69.87	55.75	62.02
FF2010	<b>67.38</b> 👍	<b>65.48</b>	76.53	<b>70.58</b>	<b>70.19</b>	<b>57.80</b>	<b>63.39</b>
WH2012	66.00	63.73	<b>77.75</b>	70.04	69.77	53.71	60.69
DErivBase	62.25	61.91	67.97	64.80	62.68	56.27	59.30
DErivBase + ZvdP2016	66.75	64.56	77.51	70.44	70.23	55.50	62.00

Table 20.2.: Results on RTE performance with prior compound splitting

### 20.6.1. Observations

The first result is that all RTE setups including prior compound splitting outperform the initial RTE performance. Based on McNemar’s test, there is only a significant improvement (👍) between INIT and FF2010, but not between INIT and ZvdP2016, or INIT and WH2012. FF2010 outperforms ZvdP2016 and WH2012. The addition of derivational information (in terms of a DErivBase lexical aligner) without prior compound splitting even harms the performance of RTE.

### 20.6.2. Discussion

The positive impact of adding compounding information can be illustrated in Example 13, which shows an entailing  $T$ - $H$  pair.

- (13) a.  $T_{\text{INIT}}$ : *Die Elektro-Lichtbogenöfen verschmelzen Metallschrott ...*  
‘The electric arc furnaces melt scrap metal ...’
- b.  $H_{\text{INIT}}$ : *Schrott wird in Elektroöfen geschmolzen*  
‘Scrap is melted in electric furnaces’
- c.  $T_{\text{ZvdP2016}}$ : *Die {Elektro Licht Bogen Ofen} verschmelzen {Metall Schrott}*  
...  
d.  $H_{\text{ZvdP2016}}$ : *Schrott wird in {Elektro Ofen} geschmolzen*

While none of the tokens in  $H_{\text{INIT}}$  can be aligned to  $T_{\text{INIT}}$  (i.e., H coverage = 0%), prior compound splitting (e.g., by ZvdP2016) leads to an H coverage of 50+%.

FF2010 clearly outperforms ZvdP2016 and WH2012.



In an error analysis, we examined all entailment classifications that were correct using FF2010 and incorrect when using ZvdP2016 or WH2012. For ZvdP2016, most errors can be attributed to **oversplitting**. Precisely, 25 out of its 37 (67.5%) misclassifications compared to FF2010 can be attributed to this problem. One reason for the **oversplitting** are cases of adjectives having a **word-inflected** form or particle/prefix verbs having a more frequent base verb. Although ZvdP2016 includes a **lemmatization** step prior to **compound splitting** (see Section 18.4.1), adjective forms like *bessere* ‘better’ are not **lemmatized** to *besser* ‘better’ (or even to *gut* ‘good’), because of the corpus lemma *bessere*, which accidentally happens to occur in the WIKIPEDIA. Instead, *bessere* is **split** into more frequent parts, leading to the analysis *be SSe Re* and thus to a lower **H coverage** for entailing cases. The prefix verb *begehen* ‘to commit’ is falsely **split** into *be + gehen* ‘be + to go’, because of the high frequency of *gehen*. Another problem of ZvdP2016 are falsely triggered **MOPs** for the **constituent normalization**. For example, the **compound** *Drogenspürhund* ‘narcotics detection dog’ is falsely analyzed to *Droge Spur Hund* ‘narcotics trace dog’, using the **word MOP** u/ü, i.e., an Umlautung which is valid for **compounds** like *Brüderbewegung* ‘Plymouth Brethren’.

For WH2012, **oversplitting** is also a major contributor of **RTE** errors, however it appears not as predominant as for ZvdP2016. 10 out of its 29 (34.5%) misclassifications compared to FF2010 can be attributed to **oversplitting**, while 4 (13.8%) misclassifications are due to **undersplitting**. One reason for the **undersplitting** is the resource-based filtering of invalid **constituent forms**, which raises the risk of missing valid **constituents**. For example, the filtering script misses the acronym *EU* (standing for *Europäische Union* ‘European Union’), even for **compounds** having a **split point marker** as in *Die Türkei verhandelt über ihren EU-Beitritt* ‘Turkey is negotiating its accession to the EU’. Another case of **undersplitting** is given in Example 14, where the year range *1890-1970* has to be **split/tokenized** for matching with the year of Charles de Gaulle’s death, 1970; establishing the correct  $T \Rightarrow H$  classification.

- (14) a.  $T_{\text{WH2012}}$ : *Charles de Gaulle, 1890-1970, französischer General...*  
           ‘Charles de Gaulle, 1890-1970, French general...’  
       b.  $T_{\text{FF2010}}$ : *Charles de Gaulle, 1890 1970, französischer General...*  
           ‘Charles de Gaulle, 1890 1970, French general...’  
       c.  $H$ : *Charles de Gaulle starb im Jahre 1970*  
           ‘Charles de Gaulle died in 1970’

The DERivBase lexical aligner does not improve the initial **RTE** performance. This is

in line with the observations made by Zeller (2016, sec. 8.2.3). There are “*overinflated*” derivational families that contain many members that are not derivationally related but only morphologically related, e.g., {*setzen* ‘to sit (sb.)’, *sitzen* ‘to sit’, *Satz* ‘set’, ...} (Zeller, 2016, p. 62). As a result, many spurious alignments are added. The positive effect of correctly added derivational alignments does not outweigh the negative impact, because there are only few *T-H* pairs in the RTE-3 dataset that contain derivational siblings in *T* and *H*, i.e., 85% of all *T-H* pairs are not relevant for DERivBase (Zeller, 2016, p. 163). In some cases, the derivational siblings have to be uncovered by performing **compound splitting**, as shown in the Example 15 presented in Zeller (2016), where *Wählern* ‘electorate’ and the **head** of *Parlaments|wahlen* ‘parliamentary elections’ are derivationally related.

- (15) a. *T*: *Chirac brauchte von den **Wählern** ein neues Mandat für seine **Regierung**...*  
 ‘Chirac needed a new mandate for his **government** from the **electorate**...’  
 b. *H*: *Parlaments**wahlen** führen zur Gründung einer neuen **Regierung** in Frankreich*  
 ‘Parliamentary **elections** create new **government** in France’

And in fact, adding both **compounding** information (e.g., prior **compound splitting** by ZvdP2016) and the DERivBase lexical aligner to the RTE system yield a slight improvement (compared to only using prior **compound splitting**), as shown in the last line of Table 20.2.

# 21. Bottom Line of Compound Splitting

This chapter constitutes the bottom line of the [compound splitting](#) Part D. We summarize all previous [compound splitting](#) chapters in Section 21.1. In Section 21.2, we conclude our work and discuss the insights, findings and results for all research questions posed in Section 15.2. Finally, in Section 21.3, we point to some further limitations of [compound splitting](#) that cannot be resolved with our contributions, and give an outlook on future work.

## 21.1. Summary

In [Chapter 15](#), we introduced [compound splitting](#) Part D. As motivation (15.1), we outlined a common statistical approach to [compound splitting](#) (15.1.1) and discussed its limitations (15.1.2). In Section 15.2 we posed some research questions that guided our work in this part and for each limitation discussed in Section 15.1.2, we proposed a contribution.

In [Chapter 16](#), we presented previous and related work on [compound splitting](#). We discussed different [splitting](#) approaches, i.e., statistical (16.1) and linguistic (16.2) approaches, and what have been the main information sources for these approaches. In Section 16.3, we compared the performance of statistical and linguistic [splitting](#) approaches, as discussed by Escartín (2014). We positioned our work as a statistical approach based on corpus frequency. One of the main contributions to [compound splitting](#) presented in this thesis is the automatic learning of [constituent inflection](#) from [word inflection](#). We compared the different knowledge resources and hand-crafted rules for modeling [constituent inflection](#) proposed by previous work. This language-independent learning of [constituent inflection](#) is aimed to be applicable to many languages. We discussed the different [target languages](#) of previous work on [compound splitting](#) and positioned our approach as [multilingual](#), i.e., as applicable to several languages (in particular Ger-

manic [closed compounding languages](#)). Another contribution proposed in this thesis is a novel [extrinsic evaluation](#) and an elaborated [intrinsic evaluation](#) method. Thus, we also discussed the different evaluation methods used in previous work.

In [Chapter 17](#), we presented the concept of the [Morphological Operation Patterns \(MOPs\)](#). We described the method of compiling [MOPs](#) given two strings  $\Sigma$  and  $\Omega$  using the Levensthein [Edit Distance \(ED\)](#) algorithm ([17.1](#)). The string pair  $(\Sigma, \Omega)$  can be derived from various sources such as a [lemmatized corpus \(word inflection\)](#) or from a [compound splitting gold standard \(17.2\)](#). The main functionality of [MOPs](#) is the application to a string  $\Sigma$  yielding the transformed string  $\Omega$ . However, the [MOP application](#) is restricted by some conventions ([17.3](#)). The application of an [MOP](#) is directed (e.g., from corpus [lemma](#) to corpus [word form](#)). For the reverse direction (e.g., from [constituent form](#) to [constituent lemma](#)), the [MOP](#) has to be inverted ([17.4](#)).

[Chapter 18](#) describes our [compound splitter](#) in detail. Firstly, the architecture of the [compound splitting](#) method was described. The main method recursively applies a [binary splitter \(18.1\)](#) on a [target compound](#) until an [atomic](#) analysis is returned. The [binary splitter](#) generates all possible [split points](#) and applies a [normalization](#) method ([18.2](#)) on the resulting [constituent forms](#). This [normalization](#) method is the **core contribution** for the [multilingual compound splitter](#). All potential [constituent forms](#) are subject to [MOP application](#) (using the inversion of learned [MOPs](#)), leading to the candidate [constituent lemmas](#). These candidates are scored using a function of [lemma](#) frequency and [MOP Suitability \(MS\)](#). Finally the  $M$  top-ranked candidates are returned. In the final step of the [binary splitter](#), all [lemma](#) combinations (including the [non-split option](#)) are scored according to a function of geometric mean of [lemma](#) scores and the [head-PoS-equality](#) feature. The highest-ranked [lemma](#) combination is subject to the recursive process ([18.3](#)). Secondly, some additional features for mitigating the impact of misleading [word inflection](#) operations ([18.4](#)) were presented: [target lemmatization](#) (using [MOP application](#)) prior to [compound splitting \(18.4.1\)](#), [compound splitting restriction to content words \(18.4.2\)](#), [PoS agreement restriction for the modifier \(18.4.3\)](#) and [lexeme agreement restriction for the head \(18.4.4\)](#). Thirdly, the different representation formats for a [compound split](#) were discussed ([18.5](#)). A [compound split](#) can be represented as a [split tree \(18.5.1\)](#), as a [lemma sequence format \(LSF\) \(18.5.2\)](#) or as a [split point format \(SPF\) \(18.5.3\)](#). Finally, experiments including the [MOP-based compound splitter](#) were conducted ([18.6](#)) for three Germanic languages: *German*, *Dutch* and *Afrikaans* ([18.6.1](#)). Before showing the experiment results, we presented and discussed all setups. As training corpus for all three languages, WIKIPEDIA has been used ([18.6.2](#)). As gold

standard for the [intrinsic evaluation](#), three German datasets were used and for Dutch and Afrikaans one test set, each ([18.6.4](#)). The preprocessing of all gold standards were described. The [intrinsic evaluation](#) metrics were presented ([18.6.5](#)). Here, we used the state-of-the-art metrics proposed by Koehn and Knight (2003), but distinguish between [split point](#) determination and [constituent normalization](#). Then, we outlined the external [compound splitting](#) methods we compared our system to ([18.6.6](#)). We grouped the experiment results into evaluation blocks ([18.6.7](#)). The first block described all proposed [compound splitting](#) features (such as prior [MOP](#)-based [lemmatization](#) of the [target compound](#)). We showed the necessity of all of them. The subsequent evaluation block concerned the comparison of the different [MOP](#) sets, i.e., the difference in performance when using [word MOPs](#) vs. [gold-constituent MOPs](#) vs. [hand-crafted constituent MOPs](#) vs. the [null-MOP](#). As a result, it turned out that [word MOPs](#) (i.e., using [word inflection](#) as approximation for [constituent inflection](#)) perform comparable to [gold-constituent MOPs](#) with respect to [split point](#) determination (SPX). In the [constituent normalization](#) discipline, [word MOPs](#) still show a solid performance (of 86+%), but cannot compete with the [gold-constituent MOPs](#). In the final evaluation block, we compared the [word MOP](#)-based [splitting](#) method (applied for *German*) with two language-specific external methods: the linguistic approach of Fritzingier and Fraser (2010) and the statistical but knowledge-rich approach of Weller and Heid (2012). The result was similar to the comparison of different [MOP](#) sets: the [multilingual splitter](#) is partially competitive with respect to [split point](#) determination but inferior in [constituent normalization](#). The main source of errors for our [splitting](#) method are misleading operations exclusively used in [word inflection](#), and cases of [undersplitting](#). Finally, the Dutch and Afrikaans version of the [multilingual splitter](#) was compared to the numbers published in Verhoeven et al. (2014). Here, our approach significantly outperforms the supervised learner used in Verhoeven et al. (2014) for *Dutch*, but we are significantly worse for *Afrikaans*. Again, the main reason for this is data sparsity due to the small corpus size.

The method for re-ranking [compound splits](#) by enriching a [splitter](#) with [Distributional Similarity \(Dsim\)](#) information was presented in **Chapter 19**. Firstly, the method was motivated: purely frequency-based [splitting](#) approaches disregard the semantic compatibility between the intended meaning of [compound](#) and [constituents](#). Then, we briefly introduced the topic of [Distributional Semantics \(DS\)](#) ([19.1](#)). The [Dsim](#) can be used as a metric for measuring the similarity between the intended meaning of [compound](#) and [constituents](#), as has been proven to be beneficial for measuring [compound](#) compositionality ([19.2](#)). Afterwards, the re-ranking method was described ([19.3](#)): starting from an

initial **compound split** ranking (19.3.1), we determine the **Dsim** between **compound** and **modifier** or **head** (19.3.2). These **Dsim** values can be combined to one score in various **Similarity Modes** (**SiModes**) (19.3.3). In the final step, these scores are multiplied with the initial **split** score; the re-ranking is based on this product (19.3.4). Cases of data sparsity undergo a special treatment (19.3.5). Then, we presented some experiments on the re-ranking method (19.4), when enriching the three German (19.4.1) **splitting** methods presented in the previous chapter (19.4.7). The **splitters** were trained on German WIKIPEDIA (19.4.2) and evaluated using the metrics **SPAcc** and **NormAcc** (19.4.3) on the HH2011GS gold standard (19.4.4). As **Distributional Semantics Model** (**DSM**) we used a model proposed by Weller et al. (2014) for measuring compositionality (19.4.5). We compared the initial ranking (**INIT**) against a re-ranking baseline with only **Dsim** information ( $RR_{DS}$ ) and against the re-ranker including the initial **split** score multiplied with the **Dsim** ( $RR_{FREQ-DS}$ ) (19.4.6). Finally, we presented the results on **compound split** re-ranking (19.4.8): as general trend we observed that the  $RR_{FREQ-DS}$  method improves the **INIT** ranking for at least one **SiMode** and the  $RR_{DS}$  baseline underperformed heavily and was significantly worse than the **INIT** ranking.

In the previous chapter, **Chapter 20**, we presented the novel **extrinsic evaluation** method for **compound splitting** using **Recognizing Textual Entailment** (**RTE**) as external **NLP** method. Firstly, we introduced (20.1) the concept of **Textual Entailment** (**TE**) (20.1.1), discussed some benefits of **RTE** for different **NLP** tasks (20.1.2) and present the lexical overlap hypothesis (20.1.3), which is the basis for the subsequent experiments. In Section 20.2, we discussed how **RTE** and **compound splitting** can go together symbiotically. There are some issues of **RTE** due to the opacity of **closed compounds** (20.2.1), which can be solved when enriching **RTE** with prior **compound splitting** (20.2.2). Conversely, **compound splitting** can benefit from **RTE** as an external method for the **extrinsic evaluation**. All kinds of **compound splitting** errors, such as **undersplitting**, **false splitting** and **oversplitting** can be penalized in **RTE** (20.2.3). While previous work used **Statistical Machine Translation** (**SMT**) for the **extrinsic evaluation** of **compound splitting**, the usage of **RTE** has several advantages over **SMT**, which were discussed in Section 20.3. For the experiments on **RTE**-based evaluation, we used a **multilingual** framework proposed by Noh et al. (2015) (20.4). Finally, we presented the experiments on the **extrinsic evaluation** using **RTE** (20.5) with a simplified algorithm (20.5.1). As training and test set, we used the **RTE-3** dataset (20.5.2) for training a supervised classifier (20.5.3). As evaluation measure, we used the **intrinsic evaluation** of **RTE** based on standard metrics (20.5.4). We considered the three German (20.5.5) **compound splitting** methods

which has already been subject in the preceding chapters (20.5.6), where the statistical approaches undergo a true-casing step prior to the inclusion into RTE (20.5.7). We additionally compared the inclusion of derivational information against the inclusion of **compounding** information (20.5.8). Finally, all results were presented and discussed (20.6): all RTE setups including **compound splitting** are superior to the initial (INIT) RTE system.

And at **the final end**, this chapter, **Chapter 21** summarizes (21.1) and concludes (21.2) the **compound splitting** part D and gives an outlook to future work (21.3).

## 21.2. Conclusion

In this section, we aim to answer the research questions posed in Section 15.2.

**RQ\_2-A:** What sources of **indirect supervision** can we use for **compound splitting**?

⇒ In Chapter 18, we propose to use **word inflection** operation as an approximation for **constituent inflection** operations. This monolingual information is based on a theory saying that German **linking elements** ‘stem from genitive and plural morphemes’ (Neef, 2009).

**RQ\_2-A-i:** How well does the approximation of using **word inflection** for **constituent inflection** work?

⇒ In general, there are only few operations for **constituent inflection** - in some languages (such as *Greek*), there is only one **compound** marker (associated with a truncation to the **word** stem). The amount of possible **word inflection** operations also depends on the language. For morphologically rich languages (such as *German*), there are more operations than for morphology-lean languages. In Section 18.6.4, we presented the various gold standards that have been used for the **intrinsic evaluation** in our experiments on **compound splitting**. For the German gold standard HH2011GS, we presented the amount of **gold-constituent MOPs** and **word MOPs** as well as the share of common **MOPs** in Table 18.4 and for VZDH2014GS in Table 18.10 (*Dutch*) and Table 18.11 (*Afrikaans*).

There are 1195 German **word MOPs** and 136 **gold-constituent MOPs**, where there are only 54 common **MOPs**. The majority of German **word MOPs** ( $\sim 95\%$ ) is not used for **constituent inflection**. These noisy **MOPs** can mislead the **compound**

splitting analysis to false constituent lemmas. In contrast, there is only a minority of 26 gold-constituent MOPs not used in word inflection ( $\sim 33\%$ ). These missing MOPs (in particular those used for Greek and Latin word stems as in the medical domain) can also lead to false constituent lemmas.

For *Dutch* and *Afrikaans*, the numbers are substantially smaller, in particular with respect to constituent inflection. We observed 908 Dutch word MOPs and 6 gold-constituent MOPs, where all gold-constituent MOPs are included in the set of word MOPs. While the majority of Dutch word MOPs ( $\sim 99\%$ ) is not used for constituent inflection, all 6 gold-constituent MOPs (= 100%) are covered by the word MOPs. Again the noisy word MOPs can interfere the compound splitting process. For *Afrikaans*, we observed 194 word MOPs (this small number also results from the very small Afrikaans training corpus) and 10 gold-constituent MOPs, where there are 8 MOPs shared by word inflection and constituent inflection. Again the majority of word MOPs ( $\sim 96\%$ ) is not used for constituent inflection, but most gold-constituent MOPs (= 80%) are covered by the word MOPs.

The impact of word MOPs on compound splitting compared to gold-constituent MOPs will be described in the answer to RQ\_2-B-i, below.

**RQ\_2-A-ii:** How expressive are the proposed MOPs?

$\Rightarrow$  MOPs are designed in a universal string-based manner such that it can be applied in any language and with any alphabet. The key elements of an MOP are the substring replacement  $\mu_i$ , the empty string  $\epsilon$ , the word beginning  $\wedge$  and the word ending  $\$$ . With these elements, it is possible to model many operations such as **prefixation** (e.g.,  $\wedge/\wedge\alpha$ ), **Umlautung** (e.g., u/ü), **suffixation** (e.g.,  $\$/\beta\$\$$ ) or **suffix replacements** (e.g.,  $\alpha\$/\beta\$\$$ ). The expressiveness of MOPs, covering these operations, is sufficient for all observed constituent inflection operations related to *German*, *Dutch* and *Afrikaans*.

**Are there morphological operations that cannot be modeled?** In the experiments presented in Section 18.6, we restricted to three Germanic languages: *German*, *Dutch* and *Afrikaans*. Here, all necessary constituent inflection operations can be modeled using MOPs. However, previous work discussed a property of Swedish **compounding**. Swedish **compounds** undergo a spelling transformation: if two constituents are to be joint such that there would be three equal consonants



in a row, one such consonant is dropped (Stymne and Holmqvist, 2008), e.g., *stopplikt* → *stop|plikt* ‘stop obligation’. While we are able to model the truncation of a word-final *p* (i.e., *p\$/\$*), the MOP application is designed context-free, i.e., we cannot check for the prerequisite condition of having three consonants in a row. The development of context-dependent MOPs will be addressed in future work.

**How ambiguous are these patterns and how is ambiguity resolved?** Most operations related to word inflection and constituent inflection occur word-final. Using the markers  $\wedge$  and  $\$$ , such operations can be applied unambiguously. However, for the word-internal operations, the concrete replacement position is underspecified. This underspecification is a deliberate feature for generalizing over various constituents. For example, the Umlautung MOP *u/ü* can be applied both to *Buch* ‘book’ (leading to *Bücher*, i.e., Umlautung at the second position) and to *Bruder* ‘brother’ (leading to *Brüder*, i.e., Umlautung at the third position). However, the underspecified replacement positions also leads to ambiguity when having several word-internal substrings that match with the source side of a replacement. For example, for the constituent *Suppenhuhn* ‘boiling hen’ (as in the compound *Suppenhühnerverkauf* ‘boiling hen sale’), the Umlautung MOP *u/ü* can be applied at the first or second *u*. By convention, this kind of ambiguity is resolved by selecting the last replacement position as default, as described in Section 17.3.

**RQ\_2-B:** How do manual-resource-lean methods compare to resource-rich and language-specific approaches?

⇒ Our manual-resource-lean compound splitter avoids hand-crafted information about constituent inflection and instead approximates this knowledge by using morphological operations learned from regular word inflection. This approximation has two weak points, as discussed in the answer for RQ\_2-A-i. Firstly, there are some operations for constituent inflection that are not covered by word inflection, and secondly, there are many operations for word inflection which are not relevant for constituent inflection. As a consequence, target compounds cannot be analyzed due to missing morphological knowledge (i.e., undersplitting) or get a false (false splitting) or too deep (oversplitting) analysis due to falsely triggered MOPs. We compared our method with gold-constituent MOPs (RQ\_2-B-i) and language-specific approaches (RQ\_2-B-ii).

**RQ\_2-B-i:** What is the difference in [splitting](#) performance when working with operations for [word inflection](#) instead of [constituent inflection](#)?

⇒ In the experimental results presented in Section 18.6.7, we used one evaluation block in which we compared the [compound splitting](#) performance using different [MOP](#) sets, including [word MOPs](#) and [gold-constituent MOPs](#). A first result was that the [splitting](#) performance using [word MOPs](#) is fairly solid for both [split point](#) determination and [constituent normalization](#). However, [word MOPs](#) are clearly inferior to [gold-constituent MOPs](#) with respect to [constituent normalization](#), while showing comparable performance to [gold-constituent MOPs](#) in [split point](#) determination. The biggest issue of [word MOPs](#) are misleading operations which are exclusively used in [word inflection](#). The proposed restrictions (e.g., [PoS](#) agreement on the [modifier](#), as described in Section 18.4.3) cannot completely eliminate the impact of misleading [word MOPs](#).

**RQ\_2-B-ii:** How competitive is the [multilingual splitting](#) approach compared to language-specific [splitting](#) methods?

⇒ In the last evaluation block presented in Section 18.6.7, we compared the [word MOP](#)-based [compound splitter](#) (ZvdP2016) with two external [splitting](#) methods: (1) the method of Fritzing and Fraser (2010), based on the morphological analyses of SMOR (FF2010) and (2) the system of Weller and Heid (2012), based on an extensive hand-crafted list of [constituent inflection](#) rules and corpus filters (WH2012).

The difference in performance for the compared [splitters](#) differ between the used gold standards. The [word MOP](#)-based [splitter](#) (ZvdP2016) shows a solid performance for all gold standards and is partially competitive (in particular with respect to [split point](#) determination) with the language-specific methods. For the [constituent normalization](#), the [multilingual splitter](#) is significantly inferior to FF2010 and WH2012.

The most frequent error our [compound splitter](#) produces with respect to [split point](#) determination is [undersplitting](#). Here, the recursive architecture has a disadvantage over the linear [splitter](#) of WH2012 in the chosen evaluation (where the [splitting](#) depth is provided by the number of gold standard [constituents](#),  $k_{gold}$ ). However, the linguistic method of FF2010 suffers even more from [undersplitting](#). So, we can conclude that this issue is not related to the [multilingual](#) aspect of our [word MOP](#)-based system.

The main source of error for **constituent normalization** is also visible when comparing the different **MOP** sets. There are misleading **word MOPs** being exclusively used in **word inflection** (e.g., in past tense inflection of finite verbs). Another typical error results from missing language-specific knowledge about **constituent inflection**, e.g., that **modifier** verbs never occur as an infinitive form.

Moreover, we compared the accuracy of **split point** determination ( $SP_{Acc}$ ) against the accuracy numbers presented in Verhoeven et al. (2014) for *Dutch* and *Afrikaans*. Here, we significantly outperform the Dutch numbers but were significantly worse for Afrikaans. The main reason for the poorer performance of the Afrikaans **word MOP**-based **splitter** is due to data sparsity. The Afrikaans training corpus is an order of magnitude smaller than for *German*, leading to many cases of **undersplitting**, where a gold **constituent** is unknown in the training data. Given the fact that *Dutch* and *Afrikaans* are similar languages, we expect to see a significant outperformance for the Afrikaans **word MOP**-based **splitter** when using a larger training corpus.

**RQ\_2-C:** How language-independent are our **splitting** approaches and what resources do they still need?

⇒ As discussed in the answer for **RQ\_2-B-ii**, the approximation of using **word inflection** as **constituent inflection** shows a solid performance for our three **target languages**: *German*, *Dutch* and *Afrikaans*. We expect that this approximation works similarly well for other Germanic **closed compounding languages** such as *Danish* or *Swedish*. While there is no need for knowledge about **constituent inflection**, our approach is still based on a monolingual corpus with **PoS**-tags and **lemmas**, two types of information that are necessary for most other **NLP** tasks.

**RQ\_2-D:** How effective is the **Dsim** information for **compound splitting**?

⇒ In Section 19.4.8, we compared the initial **split** ranking with the re-ranking using only **Dsim** information ( $RR_{DS}$ ) and using both the initial **split** score (e.g., based on corpus frequency) and in addition the **Dsim** information ( $RR_{FREQ-DS}$ ). For  $RR_{FREQ-DS}$ , we observed an improvement for all **compound splitters** in at least one **Similarity Mode** (**SiMode**).

**RQ\_2-D-i:** What is the average performance gain when adding **Dsim** information?

⇒ The performance gain differs between the inspected **compound splitters**. Since we used different test sets (which contain initial **binary split** rankings that are relevant for re-ranking), we cannot compare the performance gain across the **splitters**. For **split point** determination (SPAcc), the improvement ranges between 0.3% and 0.7% (averaged to 0.5%). The improvements are stronger for the **constituent normalization**. Here, the gain ranges between 0.5% and 1.5% (averaged to 0.9%).

**RQ\_2-D-ii:** Which frequency-based **compound splitter** benefits most from adding semantics information?

⇒ While there is only a small improvement in SPAcc for the statistical methods of WH2012 (Table 19.7) and ZvdP2016 (Table 19.8), there is a bigger improvement for the linguistic approach of FF2010 (Table 19.6). In contrast, the performance gain in NormAcc is small for FF2010 and WH2012 but largest for the manual-resource-lean method (ZvdP2016).

**How can linguistically-informed **splitting** systems be improved when adding semantics information?** The linguistic approach of FF2010 is based on the morphological analyzer SMOR and thus only morphologically plausible **compound splits** are provided. Thus, only hard cases of **splitting** ambiguity are left for resolving. One such phenomenon is the **split point ambiguity** based on **atomic** units. For example, *Punktrichter* ‘judge’, for which there are two possible readings: *Punkt | richter* ‘judge’ (lit: ‘point judge’) and *Punk | trichter* ‘punk funnel’. Another phenomenon is the **binary splitting** of 3NCs (which is comparable to **compound parsing**). For example, *Blei|kristall|glas* ‘lead crystal glass’, where a **binary splitter** has to choose between a LEFT-branched structure (*Bleikristall|glas*) and a RIGHT-branched structure (*Blei|kristallglas*). When adding semantics information, these types of ambiguity can be resolved. For example, knowing that *Punktrichter* is distributionally more similar to *Richter* than to *Trichter* promotes the correct candidate **split**.

**How does distributional similarity improve statistical **compound splitters**?** While there are only morphologically plausible candidate **compound splits** under investigation when using a linguistic **compound splitter** (e.g., FF2010), statistical **splitting** methods also propose morphologically implausible **split** options. In WH2012, there is an extensive list of morphological transformation rules modeling

**constituent inflection**. However, it is not very clear, when these rules have to be used. As a result, there are wrong **compound splits** due to a falsely triggered transformation rule. For example, the **2NC** *Zahnseide* ‘dental floss’ is falsely **split** into *Zahn(s)|eide* ‘tooth oaths’ by truncating the *s*-suffix, which is valid for **compounds** such as *Rinds* | *leder* ‘cowskin’ but invalid for the **constituent lemma** *Zahn*. Exploiting the fact that the **Dsim** between *Seide* ‘silk’ and *Zahnseide* is higher than between *Eid* ‘oath’ and *Zahnseide*, the correct **compound split** *Zahn* | *Seide* is promoted.

For the **MOP**-based **splitting** method (ZvdP2016), the morphological transformation rules are learned from **word inflection**. As discussed earlier, this can yield false **compound splits** due to misleading **word MOPs**. For example, the **word MOP** *e/a* (as in the pluralized past tense verb form of *sehen* ‘to see’: *sahen*) allows for the **compound** *Denkansatz* ‘intellectual approach’ to be **split** into *Denkan|satz* with the **constituent lemmas** *denken* ‘to think’ and *Satz* ‘sentence’. Using the knowledge that *Ansatz* is more distributionally similar to *Denkansatz* than *Satz* is, the correct **splitting** is promoted: *Denk* | *ansatz*.

Therefore, we can conclude that information about distributional similarity between **target compound** and potential **constituents** actually helps statistical **compound splitters** to mitigate the impact of lacking morphological knowledge such as which transformations are valid for **constituent inflection** and when these transformations have to be triggered.

**RQ\_2-D-iii:** What are the individual contributions of **Dsim** and corpus frequency information?

⇒ There are various reasons for the improvements in *SPAcc* and *NormAcc* across the inspected **splitters**. While the  $RR_{\text{FREQ-DS}}$  shows a consistent improvement for all inspected **compound splitters**, the  $RR_{\text{DS}}$  baseline (i.e., only **Dsim**) heavily underperforms and is even significantly worse than the **INIT** (i.e., only corpus frequency) **split** ranking. This is in line with previous work (Weller et al., 2014) and shows that isolated semantic information does not suffice but needs to be introduced as an additional feature. Therefore, we can conclude that the contribution of the corpus frequency information is much stronger than for the **Dsim** information, but using both corpus frequency and **Dsim** leads to the best performance. One example of a **compound** whose **splitting** benefits from corpus frequency is *Haarwasser* ‘hair

tonic’ with the correct and highly frequent **modifier** *Haar* ‘hair’. In contrast,  $RR_{DS}$  would select the morphologically plausible but yet unlikely and infrequent verbal **modifier** *haaren* ‘to molt’, which happens to have the higher  $Dsim$  to *Haarwasser*. On the other hand, **binary splitting** of 3NCs (i.e., **bracketing**) benefits from  $Dsim$  information. For example, using re-ranking by  $RR_{FREQ-DS}$ , the wrong **compound split** *Arbeits|platzmangel* ‘labor [lack of space]’ (top-ranked in INIT) is corrected to *Arbeitsplatz|mangel* ‘job scarcity’ in  $RR_{FREQ-DS}$ .

**RQ\_2-D-iv:** What **constituent type** provides the best-working  $Dsim$  information?

⇒ We compared the performance gain when adding  $Dsim$  information to the various **compound splitters** for both **constituent types** (SiModes): **modifier** (MOD) and **head** (HEAD). There is no clear result which **constituent type** works best. Given the fact that the **head** embodies all principle semantics of the **compound** (3.6.1), one might expect to see the highest performance gain when using the HEAD. However, this trend is not visible in the results tables 19.6 to 19.8. For FF2010, MOD is working best. In contrast, for WH2012, MOD and HEAD perform equally for SP*Acc*, whereas MOD outperforms HEAD in Norm*Acc*. And for ZvdP2016, HEAD outperforms MOD for SP*Acc* and vice versa for Norm*Acc*.

**In which cases does the **modifier** outperform the **head**?** One such case for which the  $Dsim$  between **compound** and **modifier** is more beneficial than between **compound** and **head** are 3NCs with a LEFT-branched structure (which is the majority structure (LEFT class baseline), as will be discussed in Part E) and an ambiguous **head**. For example, the 3NC *Block|flöten|spieler* ‘recorder player’, where the complex **head**, *Flötenspieler* ‘flute player’, is more distributionally similar to the **compound** than the simplex **head** *Spieler* ‘player’. In contrast, the complex **modifier** *Blockflöte* ‘recorder’ is more similar to the **compound** than the simplex **modifier** *Block* ‘block’. Therefore, the SiMode HEAD is not as beneficial as MOD.

**In which cases does the **head** outperform the **modifier**?** On the other hand, there are also LEFT-branched 3NCs, for which the **head** is not ambiguous but the complex **modifier** is infrequent (or phrase-like). For example, for the **compound** *Energie|spar|lampe* ‘energy-saving bulb’, the simplex **modifier**, *Energie*, is more similar to the **compound** than the complex (phrasal) **modifier**, *energiesparen* ‘to

save energy’. In contrast, the simplex head *Lampe* is more similar to the compound than the complex (infrequent) head, *Sparlampe*. Therefore, the SiMode MOD is not as beneficial as HEAD.

**Which type of combination of modifier and head work best?** When combining the SiModes MOD and HEAD, it turns out that the SiMode GEO outperforms both individual SiModes. This is in line with tendencies found in previous work on compositionality of compounds (Schulte im Walde et al., 2013). Moreover, we compared GEO with the two vector arithmetic SiModes: vector multiplication (MULT) and vector addition (ADD). We observed that for NormAcc, GEO outperforms both MULT and ADD, whereas for SPAcc, the performance between GEO and MULT/ADD is comparable.

**RQ\_2-E:** How suitable are the novel intrinsic and extrinsic evaluation methods for compound splitting?

⇒ As discussed in Section 15.1.2, there are various limitations of evaluation methods used for compound splitting in previous work.

For the intrinsic evaluation, a restriction to constituent lemmas disregards the fact that for some German compounds, a verbal interpretation of the modifier is as plausible as a nominal interpretation, as in *Tanz|lokal* ‘dance/dancing hall’, where both *Tanz* ‘dance’ and *tanzen* ‘to dance’ are plausible modifier lemmas.

Using RTE for the extrinsic evaluation of compound splitting is very promising. As has been discussed in Section 20.3, it has several advantages over SMT, which is most widely used in previous work about extrinsic evaluation of splitting methods. For example, there is a high agreement on TE ratings, whereas the “*general notion of quality of a translation is [...] subjective*” (Olive et al., 2011, sec. 5.1.2).

**RQ\_2-E-i:** Are there differences in the ranking of compound splitters for split point determination and constituent normalization?

⇒ In the last evaluation block of Section 18.6.7, we compared the three German compound splitters against each other, as shown in Table 18.15. Here, all splitters are compared with respect to two disciplines: (1) split point determination (measured as SPX) and (2) constituent normalization (measured as NormX). Besides the fact

that the [split point](#) determination is the easier task and shows much higher performance numbers than for the more challenging task of [constituent normalization](#), there is also a difference in the performance ranking of the [compound splitters](#). For the HH2011GS gold standard, the method of FF2010 is best in Norm $F_1$  but worst in SP $F_1$ . For the M2006GS gold standard, ZvdP2016 is worst in Norm $F_1$ , whereas it outperforms FF2010 in SP $F_1$ .

**What methods perform best with respect to split point determination?**

For the discipline of [split point](#) determination (measured in SP $F_1$ ), ZvdP2016 is best in the gold standard of HH2011GS, whereas WH2012 is best for the other two gold standards: M2006GS and HB2008GS.

**What systems are superior in constituent normalization?** For the discipline of [constituent normalization](#) (measured in Norm $F_1$ ), FF2010 is best in the HH2011GS gold standard, whereas WH2012 is best for the M2006GS gold standard. ZvdP2016 is worst in both gold standards, which is to be expected, because ZvdP2016 learns morphological operations for [constituent inflection](#) (i.e., the reverse operation of [constituent normalization](#)) automatically from [word inflection](#), introducing noisy [word MOPs](#).

**RQ\_2-E-ii:** Hoes does RTE treat the different errors occurring in [compound splitting](#): [false splitting](#), [oversplitting](#) and [undersplitting](#)?

⇒ We discussed the impact of correct and [false splitting](#) to RTE in detail in Section 20.2.3. While correct [splitting](#) improves the RTE performance (by revealing (1) [lexemes](#) that are common in the text  $T$  and the hypothesis  $H$  and (2) a large number of uncovered [lexemes](#) in  $H$ ), [undersplitting](#) does not help to overcome the limitations due to the opacity of [closed compounds](#), discussed in Section 20.2.1. In the case of [oversplitting](#) [atomic words](#) or [constituents](#) exclusively occurring in  $H$ , the number of uncovered [tokens](#) in  $H$  increases unjustified, which has a negative impact on the correct classification of entailing  $T$ - $H$  pairs. As discussed by Dyer (2009), Fritzinger and Fraser (2010) and Weller et al. (2014), phrase-based SMT is robust to [oversplitting](#), because oversplit [words](#) are often learned as phrases. Moreover, previous approaches on the [intrinsic evaluation](#) of [compound splitting](#) were not consistent (e.g., there are [splitting](#) gold standards missing non-[compounds](#) (Henrich and Hinrichs, 2011)).



The impact of **false splitting** in  $H$  is the same as for **oversplitting**: the number of **tokens** in  $H$  increases, while none matches with a **token** in  $T$ .

Actually, an error analysis for our experiment on the RTE-based evaluation described in Section 20.5 revealed that for ZvdP2016 (being compared to FF2010), most cases of misclassification ( $\frac{25}{37} \approx 67.6\%$ ) can be attributed to **oversplitting**. For WH2012 (being compared to FF2010), about 34.5% of all misclassifications are due to **oversplitting**, whereas 13.8% of all misclassifications can be attributed to **undersplitting**.

## 21.3. Future Work

In this section, we describe some limitations of the proposed **multilingual compound splitting** method, the **Dsim**-based re-ranker and the RTE-based **extrinsic evaluation** method presented in the previous chapters.

### 21.3.1. Multilingual Compound Splitting

#### Non-binary Split Tree Structure

The **compound splitter** presented in Chapter 18 is limited to a **binary structural analysis** of **closed compounds**. Except for the special case of **compounds** containing **split point markers** (e.g., hyphens), each **constituent** resulting from a **binary split** needs to be known (i.e., the used set of **MOPs** can **normalize** it to a **lemma** contained in the **Lemma Resource (LR)**). The longer the **target compound**, the lower the chance of finding two composed **constituent lemmas** within the **LR**.

Besides unknown **constituents**, **phrasal compounds** (3.7.3) pose a big challenge for **binary compound splitters**. For example, the **compound** *Langsamfahrstelle* ‘temporary speed restriction’ cannot be **binary split**, because the **modifier** *Langsam fahren* ‘to drive slowly’ is not a lexical (one-word) unit but a phrase.

The above-mentioned issues can be addressed by extending the **binary splitter**, described in Section 18.1, to a flexible **Nary splitter** that starts with a **binary splitting** step ( $N = 2$ ) and falls back to a **ternary splitting** step ( $N = 3$ ) if no **split** can be found. For the example of *Langsamfahrstelle*, a **ternary splitting** would yield plausible (flat) tree structure as shown in Figure 21.1.

## 21. Bottom Line of Compound Splitting

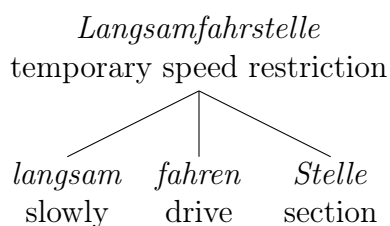


Figure 21.1.: Example of a ternary split tree structure for *Langsamfahrstelle*

### Context-dependent MOPs

As described in Section 16.1.1, Swedish **compounds** undergo a spelling transformation: if two **constituents** are to be joint such that there would be three equal consonants in a row, one such consonant is dropped (Stymne and Holmqvist, 2008). For modelling this behaviour, a third consonant is allowed if a **split point** separates two consecutive consonants (e.g., *stop|plikt* → *stop**p** plikt* ‘stop obligation’).

The presented **MOPs** are applied out of context. While a truncation of a trailing *p* could be represented as **p\$/**, it cannot be restricted only to cases where three consonants in a row are joint.

For this task, the framework of **MOPs** has to be extended by a look-behind and look-ahead operator. For example,  $\langle p_1, \text{p}/, p_2 \rangle$ , where  $p_1$  refers to the string that directly precedes the source side of a substring replacement  $\mu_i$  and  $p_2$  refers to the string that directly succeeds the source side of  $\mu_i$ . While the lookbehind could be implemented without any changes in the proposed **compound splitting** architecture, the lookahead operation seems to be more difficult, because **constituent normalization** (which is usually a **word-final** operation) is currently performed in isolation. A solution would be an additional feature for the combination model presented in Section 18.3.

### Derivational Information for Modifier Agreement

As described in Section 18.4.5, the **PoS-agreement** restriction is **too lenient** for filtering noisy **word MOPs**. For example, the **word MOP** **a/ä:\$/er/** (which is valid for **nouns** like *Mann* ‘man’) can be falsely triggered for the **compound** *Läufer**te**am* ‘runner’s team’, leading to the nominal **modifier lemma** *Lauf* ‘run’. However, applying the **lexeme** agreement restriction to the **modifier** would be **too restrictive**, because there are many **lexemes** that do not share **MOPs** from **word inflection** and **constituent inflection**, e.g., nouns ending on *-heit* (e.g., *Kindheit* ‘childhood’: while *Kindheit* gets *s*-suffixed as a **modifier** (as in *Kindheits|erinnerung* ‘childhood memory’), *Kindheits* is not a paradigmatic **word form**). Therefore, the **MOP application** has to generalize over the **word** category.

A possible solution for this could be the inclusion of derivational information (i.e., *Lauf* → *Läufer* ‘run → runner’) automatically derived from [Distributional Semantics \(DS\)](#) (e.g., [DERivBase](#) (Zeller et al., 2013)). If the [word inflection-based MOP application](#) would result in a derivation (rather than [constituent inflection](#)) of a [constituent lemma](#), the [MOP application](#) would be prohibited.

### 21.3.2. Shallow Semantics Support

#### Constituent-wise Distributional Similarity

In Chapter 19, we presented a [compound splitting](#) enrichment by re-ranking [split](#) options using the [Dsim](#) between the intended meaning of [compound](#) and [constituent](#) (e.g.,  $\cos(\overrightarrow{Eidotter}, \overrightarrow{Dotter})$  vs.  $\cos(\overrightarrow{Eidotter}, \overrightarrow{Otter})$ ). While the enrichment with this type of [Dsim](#) significantly improves frequency-based [compound splitting](#) approaches, it is limited by the fact that the [compound](#) has to provide enough corpus evidence for being representative in a [DSM](#).

An alternative way is the [Dsim](#) between the [constituents](#), e.g.,  $\cos(\overrightarrow{Ei}, \overrightarrow{Dotter})$  vs.  $\cos(\overrightarrow{Eid}, \overrightarrow{Otter})$ .

#### N-ary Distributional Similarity

The formulas for the various [Similarity Modes \(SiModes\)](#), presented in Section 19.3.3 are designed for  $N$ -ary [compounds](#). While we have shown the positive impact of re-ranking [binary](#) splits, we will investigate the performance of  $N$ -ary splits ( $N > 2$ ): how does the performance gain correlate with respect to the number of [constituents](#)? As shown in Section 18.6 for the M2006GS gold standard, [splitting compounds](#) with three or more [constituents](#) is harder. Thus, the re-ranking approach is expected to be even more beneficial for those [compounds](#).

### 21.3.3. Evaluation Method

#### Split Tree Evaluation

The current [intrinsic evaluation](#) method presented in Section 18.6.5 is restricted to assess agreement with the information provided in the gold standards, i.e., it looks for matching [SPFs](#) and [LSFs](#). However, the recursive architecture of our [compound splitter](#) presented in Figure 18.1 produces a hierarchical output of [binary splitting](#) decisions, i.e., a [binary split tree](#). To the best of our knowledge, there are no [compound splitting](#) gold standards that provide a hierarchical structure.

We will develop a hierarchical version of the HH2011GS gold standard with a **binary-split closure** on of HH2011GS, i.e., by recursively replacing a **constituent** with its **sub-constituents** as long as they can be found within HH2011GS. This leads to an approximation of **split trees**, which is inspected by human annotators for correctness and completeness.

### The Split Point Level

The **intrinsic evaluation** method presented in Section 18.6.5 is based on the entire **compound**, i.e., the **compound split** is judged as either correct or incorrect. The evaluation category **wrong faulty split** (**wf**) proposed by Koehn and Knight (2003) subsumes cases like **oversplitting/undersplitting** while hitting one gold **split point**, and splitting without hitting any gold **split point**. This evaluation on the **compound** level is not as fair as evaluating on the **split point** level. For example, the German **compound** *Raubkopierer* ‘software pirate’ is correctly **split** into *Raub | kopierer*. While on the **compound** level, the **compound splits** *Raub | kopier | er* and *Raubkopier | er* are treated equally (as false), the first analysis should be considered better than the second, because it hits one correct **split point**. Using an evaluation on a **split point** level, provides a reward for partially correct **compound splits**. Moreover, an **intrinsic evaluation** based on a **split point** level is also beneficial for **SMT**, which can already profit from a partially correct **compound split**.

### Combining Compound Splitting with other Lexical Resources in RTE

In Section 20.2.2, we described that the impact of correct **compound splitting** on **RTE** is limited, as shown in Example 10 and Example 11, repeated in Example 16 and Example 17.

- (16) a. *T: Der Pilot fliegt in einem Jet* ‘The pilot controls a jet’  
 b. *H: Der {Flug Kapitän} fliegt ein {Flug Zeug}*  
 ‘The aircraft captain controls an airplane’
- (17) a. *T: Peter fährt einen Mercedes* ‘Peter drives a Mercedes’  
 b. *H: Peter ist ein {Auto Fahrer}* ‘Peter is a car driver’

In Example 16, the **atomic word** *Pilot* is synonymous to the **compound** *Flugkapitän*. When integrating semantic knowledge **prior** to **compound splitting**, the entailing relation in Example 10 could be revealed. In Example 17, the **modifier** *Auto* in *H* is a hypernym of *Mercedes* in *T* and the **head** *Fahrer* in *H* is a derivation of *fährt* (or *fahren*) in *T*.

When integrating semantic and derivational knowledge **after** compound splitting, the entailing relation between  $T$  and  $H$  can be determined. The integration of knowledge derived from lexical resources is not trivial, because it needs to be used both *prior to* and *after* compound splitting.

A possible solution could be a double-check. If an alignment between **lexemes** in  $T$  and  $H$  using several lexical resources fails, it is subject to **compound splitting** and a downstream lookup with these lexical resources.

### Evaluation of Dutch Compound Splitting on RTE

While there are only **RTE** gold standards available for *German* as the only **closed compounding language**, as discussed in Section 20.5.5, the Cross-Language Evaluation Forum (CLEF) provides Dutch data for the similar task of Answer Validation Exercise<sup>1</sup> (AVE), which uses the same **RTE** formalism for pushing **QA** (Bikel and Zitouni, 2012, sec. 6.2.5). We will investigate the usability of AVE data for **RTE**-based evaluation of **compound splitting**: e.g., we aim to figure out, how different is AVE from **RTE**?

---

<sup>1</sup><http://nlp.uned.es/clef-qa/ave/>

*21. Bottom Line of Compound Splitting*

Part E.

## Compound Parsing





## 22. Introduction to Compound Parsing

In this part, we present and elaborate the work published in Ziering and Van der Plas (2014), Ziering and Van der Plas (2015a) and Ziering and Van der Plas (2015b).

In Section 3.6.4, we discussed that **compounds** that have more than two **constituents** are syntactically ambiguous with respect to a **binary** structure. We generally assume **binary** syntactic structures for linguistic expressions such as **compounds**. In the case of a **ternary** structure a **3NC** would not be syntactically ambiguous. The assumption of **binary** structures is in line with common phrase structure grammars for sentence parsing, as illustrated in Figure 22.1.

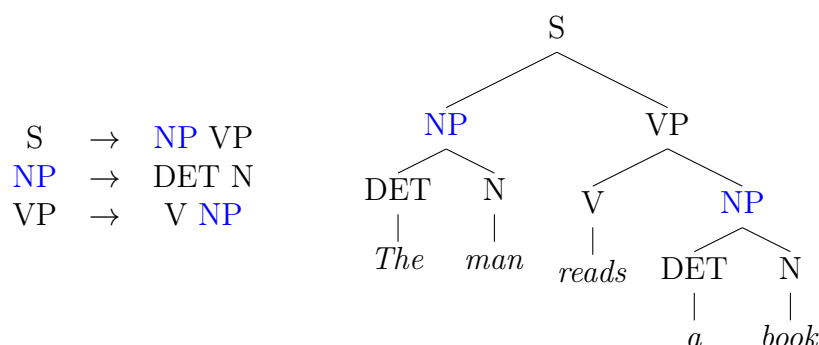


Figure 22.1.: Example of binary structure for sentence parsing

In many cases, the intended meaning of a **compound** correlates with its syntactic structure, as exemplified for the **ternary nominal compound** *natural language processing* in Figure 3.4 in Section 3.8.3, repeated in Figure 22.2. A **RIGHT**-branched structure of *natural language processing* means the natural processing of any language (e.g., the cerebral processing of a programming language), whereas a **LEFT**-branched structure reflects the common understanding of **NLP** as the (machine-based) processing of natural languages.



Figure 22.2.: Tree structures for ‘Natural Language Processing’

## 22.1. Motivation

### 22.1.1. The Importance of Compound Parsing

For tasks including [Natural Language Understanding \(NLU\)](#) such as [Machine Translation \(MT\)](#), there is need for knowing the [internal structure](#) of [compounds](#) to be processed. For the task of semantically interpreting more complex [compounds](#) (e.g., determining the implicit [semantic relation](#) holding between [modifier](#) and [head](#)), it is necessary to know which group of [atomic constituents](#) forms the [modifier](#) and [head](#) (i.e., the [immediate constituents](#)).

Moreover, “for [compounds](#) longer than two [words](#), the correct pronunciation also depends on their [internal syntactic structure](#), which makes a [noun compound parser](#) an indispensable component of an ideal speech synthesis system” (Nakov, 2013).

Despite the importance of [parsing compounds](#) and [NPs](#), this task is an “understudied language analysis problem” (Nakov and Hearst, 2005). Most sentence parsers neglect analyzing [base NPs](#), since the “main training corpus for parsers, the [Penn Treebank \(PTB\)](#) (Marcus et al., 1993) leaves a flat structure for [base NPs](#). Recent annotations by Vadas and Curran (2007a) added [NP](#) structure to the [PTB](#).” (Pitler et al., 2010).

### 22.1.2. Behaghel’s First Law - our Guiding Principle

#### Semantic Association for Compound Parsing

The core idea of [compound parsing](#) is to (recursively) group [constituents](#) that *belong* together, starting with the strongest *association*. In contrast to grammar-based syntactic parsing of sentences (as illustrated in Figure 22.1), [compound parsing](#) is semantically motivated. Thus, the semantically motivated association between [immediate constituents](#) is subsequently called [semantic association](#).

### Behaghel’s First Law

Behaghel (1909) described several linguistic principles about the position of **atomic** and complex expressions within a sentence. His First Law says

*Elements that belong close together intellectually will also be placed close together*

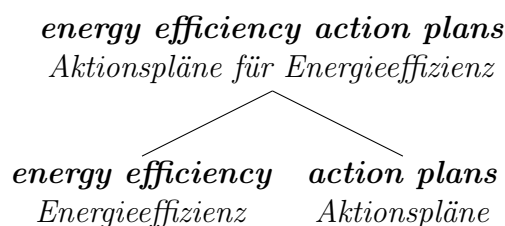
According to Behaghel’s First Law, there is a correlation between spatial proximity within a sentence and **semantic association** (where we presume that **semantic association** measures the degree of belonging together intellectually).

### Behaghel’s First Law for Monolingual Compound Parsing

Nakov and Hearst (2005) defined various features for monolingually **parsing** a **3NC A B C**. They defined *surface features* such as dashes (e.g., ‘A-B’) or optional **closed compounding** (e.g., ‘AB’), as well as *paraphrase features* such as *prepositional phrases* (e.g., ‘C from the A B’), *copula paraphrases* (e.g., B C *that/which is a* A) and *verbal paraphrases* (e.g., C *associated with* A B). Although Nakov and Hearst (2005) did not refer to Behaghel’s First Law or to the **word distance** between **constituents**, all of these features exploit a smaller **word distance** between the **constituents** with the strongest **semantic association**, i.e., those **constituents** that need to be merged during **parsing**. Since the features defined by Nakov and Hearst (2005) rely on any (monolingual) corpus occurrence of the **types** of each **constituent** A, B and C, their approach cannot handle structural ambiguity depending on context, as exemplified above with Figure 22.2.

### Behaghel’s First Law for Cross-lingual Compound Parsing

As discussed in Chapter 5, during our exploration of **compounding** across languages, as found in **parallel corpora**, we observed that there are various ways how an English **compound** can be realized in other languages. Some ways, such as **phrasal equivalents** or **aligned phrases** (Section 5.2), also reveal a variation in the **word distance** between the **constituent equivalents**. For example, the **4NC** *energy<sub>A</sub> efficiency<sub>B</sub> action<sub>C</sub> plans<sub>D</sub>* can be paraphrased in *German* as *Aktionspläne<sub>CD</sub> für Energieeffizienz<sub>AB</sub>* (lit: ‘action plans for energy efficiency’). The German **phrasal equivalent** groups AB and CD (as **closed compounds**) spatially separated by the preposition *für*, leading to the **parse tree** shown in Figure 5.1, repeated in Figure 22.3.

Figure 22.3.: Balanced tree structure for *energy efficiency action plans*

An additional benefit of [cross-lingual](#) evidence from [parallel corpora](#) (rather than from bilingual dictionaries) as indicator for [semantic association](#) (and therewith for [compound parsing](#)) is that a [compound](#)'s [cross-lingual equivalent](#) is **token-based** and caters for context-dependent structural ambiguity.

## 22.2. Contributions and related Research Questions

In this thesis, we present several [cross-lingual compound parsing](#) methods with which we can provide the following contributions. Moreover, in this section, we repeat and refine some research questions posed in Section 1.3 and add some additional ones. The enumeration of the following research subquestions for [compound parsing](#) continues the enumeration starting for the previous research subquestions for [compound splitting](#) posed in Section 15.2.

We aim to answer the following research question.

**RQ\_2-A:** What sources of [indirect supervision](#) can we use for [compound parsing](#)?

### 22.2.1. Spatial Proximity for Semantic Association

The features defined by Nakov and Hearst (2005) are based on fixed (wildcard) paraphrases and exploit the same guiding principle (22.1.2) but not explicitly. Instead, our [compound parsing](#) methods presented in this thesis are directly relying on the spatial proximity (e.g., in terms of [word distance](#)) as a measure for [semantic association](#). We illustrate the direct usefulness of the First Law of Behaghel (1909) on the task of [compound parsing](#).

We aim to answer the following research subquestion.

**RQ\_2-A-iii:** What potential does our guiding principle (22.1.2) have for [cross-lingual compound parsing](#)?

### 22.2.2. Cross-lingual Perspective for Token-based Parsing

As will be discussed in Chapter 23, all previous work on [compound parsing](#) is monolingual and for the larger part processes [compounds type](#)-based rather than [token](#)-based. Exceptions include Vadas and Curran (2007b) or Pitler et al. (2010), who developed supervised [NP parsers](#) that rely on manual annotations (Vadas and Curran, 2007a). As a consequence, most previous approaches neglect the phenomenon of structural ambiguity, where there are several plausible syntactic structures depending on the context (or the intended meaning). While it is hard to find resources that provide expressive paraphrases for [compounds](#) in monolingual context, we show that taking a [cross-lingual](#) perspective (as given by a [parallel corpus](#)) allows for finding expressive [phrasal equivalents](#) for a [compound token](#) (i.e., a context-dependent instance of a [compound](#)), serving for [token](#)-based [compound parsing](#).

We aim to answer the following research subquestion.

**RQ\_2-A-iv:** Does the [token](#)-based approach, provided by the [cross-lingual](#) perspective, lead to a better [parsing](#) performance?

### 22.2.3. Simple Metric for Cross-lingual Spatial Proximity

We propose a novel and simple metric for measuring the spatial proximity between the [constituents](#) of a [target compound](#) within the [cross-lingual](#) perspective. More specifically, we define a metric that reflects the [word distance](#) of the [constituent equivalents](#) within a [cross-lingually](#) aligned sentence. We will call this metric the [aligned word distance](#) ([AWD](#)). More details about this metric will follow in Section 25.1.

We aim to answer the following research subquestions.

**RQ\_2-A-v:** How useful is the proposed [AWD](#) metric?

**RQ\_2-A-vi:** How competitive is [cross-lingual compound parsing](#) compared to other knowledge-lean [parsers](#)?

### 22.2.4. Automatic Detection of Semantic Indeterminacy

As described in Section 3.8.3, [compounds](#) for which different [internal structures](#) correlate with the same intended meaning are called [semantically indeterminate](#). We exploit the fact that a [semantically indeterminate compound](#)  $\Psi$  happens to be translated to [phrasal equivalents](#) revealing different [internal structures](#) of  $\Psi$  (with respect to different spatial

proximities of the [constituent equivalents](#)) in different [support languages](#), as discussed in Section 5.4.4. Accumulating the [parse trees](#) derived from different [phrasal equivalents](#) allows us to identify cases of [semantic indeterminacy](#) (i.e., when there are several [parse trees](#) having the same (or similar) frequency).

We aim to answer the following research subquestion.

**RQ\_2-A-vii:** How precise is the [cross-lingual](#) detection of [semantic indeterminacy](#)?

## 22.3. Outline

The [parsing](#) Part E of this thesis has the following structure.

A detailed discussion on previous approaches to [compound parsing](#) is given in **Chapter 23**. In **Chapter 24**, we will start our journey with a **pilot study** for [cross-lingual compound parsing](#) that is based on a pattern-based approach. These patterns are motivated by frequent [aligned phrases](#) (as observed in the [XCI](#) in Section 10.2) and are thus called [Aligned Phrase Pattern \(APP\)](#). The main chapter of this part, **Chapter 25**, presents various methods for [compound parsing](#) that are based on the [cross-lingual metric aligned word distance \(AWD\)](#). Both deterministic (as in Section 25.2) and non-deterministic approaches (as in Section 25.3) are proposed and compared within different experiments. The [compound parsing](#) Part E is summarized (26.1) and concluded (26.2) in **Chapter 26**. Finally, we give an outlook on future work (26.3).

## 23. Related Work on Compound Parsing

In this chapter, we present an outline of previous and related work on the subject of this part, i.e., the [cross-lingual parsing](#) of [compounds](#).

In the description of each [parsing](#) approach, we focus on **five features**:

1. **Compound class** - is the [parser](#) able to determine the structure of a specific kind of [compound](#) (e.g., only the binary LEFT/RIGHT classification of [3NCs](#); or just any sequence of nouns, i.e., [kNCs](#); [closed](#) or [open compounds](#)) or is it applicable for structuring any kind of phrase (e.g., [noun phrases \(NPs\)](#))?

Although our experiments are designed for [parsing 3NCs](#) and [4NCs](#), our [cross-lingual compound parser](#) is applicable to any expression providing a sequence of [content words](#) as [constituents](#), where it does not matter which category the [constituents](#) have. For example, the RIGHT-branched NP *big<sub>A</sub> price<sub>B</sub> label<sub>C</sub>* can be [parsed](#) using the German NP *großes<sub>A</sub> Preisschild<sub>BC</sub>*.

The [cross-lingual compound parsing](#) methods presented in this part are designed for [open](#) constructions. This thesis also provides methods for determining the [internal structure](#) of [closed compounds](#) within the task of [compound splitting](#), outlined in Part D.

2. **Language** - for what [target languages](#) is the [parsing](#) method designed and what [support languages](#) can be used?

While the experiments evaluating the performance of our [cross-lingual compound parser](#) restricts to English [target compounds](#), the [parser](#) is designed to be language-independent and can be applied to any [open compounding language](#) occurring in a [parallel corpus](#).

As aligned [support languages](#), our method can use any language that provides an expressive paraphrase, revealing a difference in [semantic association](#) between the adjacent [target constituents](#).

3. **Contextuality** - can the [parser](#) be applied to [compound tokens](#) or only to [types](#), i.e., does a [compound](#) get different structures depending on the context (e.g., while *[online music] service* refers to a service for online music, *online [music service]* means an online service for music which possibly delivers online ordered music by mail)?

Our proposed [cross-lingual compound parsing](#) methods are designed [token](#)-based and thus take into account all context-dependent structural ambiguity. By accumulating [parse trees](#), it is possible to combine the results from several [compound](#) instances and also provide a [type](#)-based [parsing](#) result.

4. **Supervision** - is the method supervised (i.e., based on training data containing [parsed compounds](#)) or unsupervised (e.g., based on bigram corpus frequency)?

Our [cross-lingual compound parser](#) is unsupervised as much as it does not rely on [parsed](#) training data. However, [cross-lingual compound parsing](#) exploits the [cross-lingual](#) information about the [internal structure](#) (in terms of aligned phrases) provided with [parallel corpora](#). As discussed in the introduction of the thesis (Chapter 1), this kind of [indirect supervision](#) is called [cross-lingual supervision](#).

5. **Method and features** - which algorithm (e.g., a machine learning-based approach) is used with what features (e.g., manually defined rules or patterns or corpus frequency) for [parsing compounds](#)?

While the rule-based [Aligned Phrase Pattern Parsing \(APPP\)](#) is based on a set of manually defined [Aligned Phrase Patterns \(APPs\)](#), the methods which will be presented in Chapter 25 rely on the fully automatic [aligned word distance \(AWD\)](#) metric.

For structuring this chapter, we group previous work with respect to the [compound](#) class. But first, we describe some basic approaches (23.1) and common [Association Measure \(AM\)](#) for [compound parsing](#) (23.2).

## 23.1. Basic Approaches to Compound Parsing

When overlooking previous work on [compound parsing](#), outlined below, we can differentiate between two basic unsupervised approaches: the [adjacency model \(AdjMod\)](#) and the [dependency model \(DepMod\)](#) - “most approaches to the problem use unsupervised methods, based on competing [association](#) strength between two of the [words](#) in the [compound](#)” (Vadas and Curran, 2007a).



There are two reasons for a RIGHT-branched 3NC A B C: (1) B C form a (possibly non-compositional) **compound**, e.g., as in *home [health care]*, and (2) A and B independently modify C, e.g., as in *adult [male rat]* (Nakov and Hearst, 2005). Each reason corresponds to one of the two basic approaches, described below.

The following discussion on these models is partly borrowed from Nakov and Hearst (2005) and Nakov (2013).

### 23.1.1. Adjacency Model

The earliest statistical approaches for **parsing** the most frequent complex **compound** class, 3NC, reach back to the early eighties, where Marcus (1980) developed the so-called **adjacency model** (**AdjMod**), which helps for **parsing ternary compounds** (TCs) (i.e., **compounds** with three **atomic constituents** which are not necessarily nouns) by comparing the **semantic association** strengths between the adjacent **constituents**.

The **AdjMod** considers the first reason for a RIGHT-branching structure: B C forms a (possibly non-compositional) **compound**.

For a TC A B C, the **AdjMod** decides whether B is more associated to A (leading to a LEFT-branched structure) or to C (leading to a RIGHT-branched structure). There are different ways of measuring the strength of **semantic association**. For the statistical approaches, statistical **Association Measures** (**AMs**) between A and B are compared with **AMs** between B and C. A discussion on possible **AMs** is given in Section 23.2.

We can define the **AdjMod** as a function mapping the classes LEFT, RIGHT or UNKNOWN to a TC A B C as follows:

$$\text{AdjMod}(A B C) = \begin{cases} \text{LEFT} & \text{if } AM(A, B) > AM(B, C) \\ \text{UNKNOWN} & \text{if } AM(A, B) = AM(B, C) \\ \text{RIGHT} & \text{if } AM(A, B) < AM(B, C) \end{cases} \quad (23.1)$$

### 23.1.2. Dependency Model

An alternative to the **AdjMod** for **parsing TCs** is the so-called **dependency model** (**DepMod**), initially proposed by Lauer (1994). This syntactically motivated model compares the **semantic association** between the dependent **constituents**.

The **DepMod** considers the second reason for a RIGHT-branching structure: both A and B modify C.

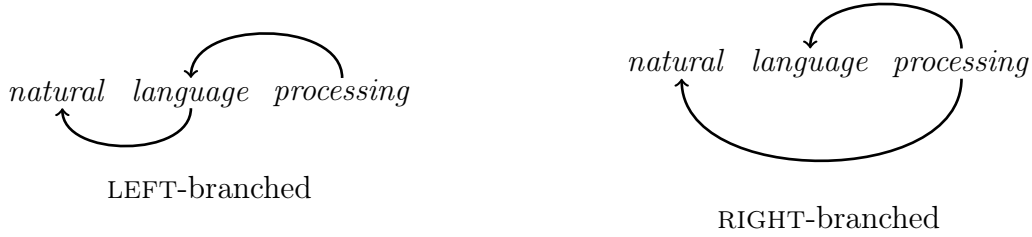


Figure 23.1.: Dependency relations for LEFT- and RIGHT-branched *natural language processing*

Figure 23.1 shows the two syntactic dependency relations for a LEFT- and RIGHT-branched analysis of *natural language processing*, where the arcs point from the **heads** to the **modifiers** (Nakov, 2013). Both the LEFT- and RIGHT-branched structures have a dependency relation between C and B, i.e., it is the processing of languages. The structures differ in the **head** that points to the leftmost **modifier**, *natural*. While in the LEFT-branched structure, *language* rules *natural*, in the RIGHT-branched structure, it is *processing*. In other words, a RIGHT-branched analysis has two **atomic modifiers**.

We can define the **DepMod** as a function mapping the classes LEFT, RIGHT or UNKNOWN to a **TC A B C** as follows:

$$\text{DepMod}(A B C) = \begin{cases} \text{LEFT} & \text{if } AM(A, B) > AM(A, C) \\ \text{UNKNOWN} & \text{if } AM(A, B) = AM(A, C) \\ \text{RIGHT} & \text{if } AM(A, B) < AM(A, C) \end{cases} \quad (23.2)$$

The fact that the RIGHT-branched structure has two dependent **atomic modifiers** for the rightmost **head** means that there is no difference in the dependency structure when swapping the two **atomic modifiers**, as shown in Figure 23.2.



Figure 23.2.: Dependency relations for swapped modifiers in a RIGHT-branched **TC**

As discussed above, a crucial difference between the **adjacency model** and the **dependency model** is the perspective of a RIGHT-branched analysis. While the **AdjMod** checks

whether  $B\ C$  is a **compound** (i.e., a test for lexicalization), the **DepMod** checks for a dependency relation between  $C$  and  $A$  (i.e., a test for syntactic modification).

This means that the **DepMod** cannot be applied to **TCs**  $A\ B\ C$  if  $B\ C$  is lexicalized/non-compositional. The fact that there is a strong **semantic association** between  $A$  and  $B\ C$  does not necessarily mean that there is a strong **semantic association** between  $A$  and  $C$ . Thus, statistical approaches that count the occurrence of  $A\ C$  as adjacent **words** is misleading. An alternative information might be the frequency of dependency relations between  $A$  and  $C$  in a dependency-parsed training set, or the frequency of instances of the pattern  $A\ c_i\ C$  (for any valid **constituent**  $c_i$ , e.g., a noun). We will investigate the performance of these alternative dependency models in future work (see Section 26.3.5).

### 23.1.3. Hybrid Adjacency-Dependency Model

There are various ways of combining the two models. In the **APPP<sub>WA</sub>** method of our pilot study (Section 24.3), we integrated both models: we voted for **RIGHT** if the **target constituent** sets  $\{B, C\}$  or  $\{A, C\}$  are aligned to the complex unit of the underlying **APP**. Another hybrid model in **cross-lingual compound parsing** based on **AWD** tree annotation is suggested as future work in Section 26.3.4.

Alternative hybrid models of previous work will be discussed below.

## 23.2. Association Measures

While presenting an exhaustive description of all **Association Measures (AMs)** used in **NLP** would extend the scope of this thesis, we would like to outline a few **AMs** that have previously be used in **compound parsing**. This survey is borrowed from Nakov and Hearst (2005). The disjunction  $(A|B)$  points to the two possible approaches: **AdjMod** ( $B$ ) and **DepMod** ( $A$ ).

### Bigram Frequency

The most simple and straightforward way is to compare the plain bigram corpus frequency between the relevant **word** pair, given in Formula 23.3.

$$\text{parse}(A\ B\ C) = \begin{cases} \text{LEFT} & \text{if } \text{freq}(A, B) > \text{freq}((A|B), C) \\ \text{UNKNOWN} & \text{if } \text{freq}(A, B) = \text{freq}((A|B), C) \\ \text{RIGHT} & \text{if } \text{freq}(A, B) < \text{freq}((A|B), C) \end{cases} \quad (23.3)$$

### Probability

If  $Pr(w_i \rightarrow w_j | w_j)$  is considered as the probability that the word  $w_i$  precedes a given fixed word  $w_j$ , and assuming that the distinct [head-modifier](#) relations are independent, we can define the following probabilities as given in Formula 23.4.

$$\begin{aligned}
 Pr(\text{RIGHT}) &= Pr((A|B) \rightarrow C|C) \\
 Pr(\text{LEFT}) &= Pr(A \rightarrow B|B) \\
 \text{parse}(A B C) &= \begin{cases} \text{LEFT} & \text{if } Pr(\text{LEFT}) > Pr(\text{RIGHT}) \\ \text{UNKNOWN} & \text{if } Pr(\text{LEFT}) = Pr(\text{RIGHT}) \\ \text{RIGHT} & \text{if } Pr(\text{LEFT}) < Pr(\text{RIGHT}) \end{cases} \quad (23.4)
 \end{aligned}$$

### Chi Squared ( $\chi^2$ )

For measuring  $\chi^2$  between two [words](#)  $w_i$  and  $w_j$ , the following information is necessary:

$\alpha$ :  $freq(w_i w_j)$  frequency of bigrams starting with  $w_i$  and ending on  $w_j$

$\beta$ :  $freq(w_i \overline{w_j})$  frequency of bigrams starting with  $w_i$  but **not** ending on  $w_j$

$\gamma$ :  $freq(\overline{w_i} w_j)$  frequency of bigrams **not** starting with  $w_i$  but ending on  $w_j$

$\delta$ :  $freq(\overline{w_i} \overline{w_j})$  frequency of bigrams **neither** starting with  $w_i$  **nor** ending on  $w_j$

$N$  total number of bigrams in the underlying corpus

The  $\chi^2$  metric is then computed using Formula 23.5.

$$\chi^2(w_i, w_j) = \frac{N \cdot (\alpha \cdot \delta - \beta \cdot \gamma)^2}{(\alpha + \gamma) \cdot (\beta + \delta) \cdot (\alpha + \beta) \cdot (\gamma + \delta)} \quad (23.5)$$

The [parsing](#) decision works straightforward as given in Formula 23.6.

$$\text{parse}(A B C) = \begin{cases} \text{LEFT} & \text{if } \chi^2(A, B) > \chi^2((A|B), C) \\ \text{UNKNOWN} & \text{if } \chi^2(A, B) = \chi^2((A|B), C) \\ \text{RIGHT} & \text{if } \chi^2(A, B) < \chi^2((A|B), C) \end{cases} \quad (23.6)$$

### 23.3. Parsing of Noun Compounds

“Most [compound bracketing](#) research has focused on [three-Noun Compounds](#)” (Barrière and Ménard, 2014).

**Marcus (1980)** was the first who described a simple method for resolving the [3NC bracketing](#) ambiguity. He proposed an unsupervised [parser](#) for English out-of-context [3NCs](#),  $A B C$ , based on three rules:

- if  $[A,B]$  is semantically implausible, select  $[B,C]$  as [immediate constituent](#), and vice versa for  $[B,C]$
- else if  $[B,C]$  is semantically more plausible than  $[A,B]$ , select  $[B,C]$  as [immediate constituent](#)
- else select  $[A,B]$  as [immediate constituent](#)

This [adjacency model](#) ([AdjMod](#)) has been adopted in various subsequent approaches.

As discussed by Lauer (1995a), **Pustejovsky et al. (1993)** is one of the first to build an empirical method based on the [AdjMod](#). For a given English out-of-context [3NC](#), both possible [bracketed word](#) pairs are inspected for corpus evidence. The [word](#) pair that has corpus evidence is chosen as [immediate constituent](#). Pustejovsky et al. (1993) do not cover cases where no [word](#) pair has evidence or both have.

In contrast, these cases are covered by **Liberman and Sproat (1992)**, who developed a more elaborated approach for English out-of-context [3NCs](#),  $A B C$ , in which the [Mutual Information](#) ([MI](#)) between  $A$  and  $B$  is compared to the [MI](#) between  $B$  and  $C$ . The [word](#) pair with the highest [MI](#) is chosen as [immediate constituent](#).

A more sophisticated implementation of the [AdjMod](#) is presented by **Resnik (1993)**. He defined a *selectional association* between a predicate and a [word](#) as the contribution of the word to the conditional entropy of the predicate. This association is computed for each possible [word](#) pair in the [compound](#), where one [word](#) is the predicate and the other [word](#) is the argument, and the [word](#) pair with the highest selectional association is used as [immediate constituent](#) (Lauer, 1995a). The values for the selectional association are estimated from the parsed WSJ. Testing the performance on 160 [3NCs](#), the approach of Resnik (1993) achieves an accuracy of 73%, outperforming the LEFT-class baseline (64%) (Lauer, 1995a).

**Lauer (1994)** was the first to switch from the [AdjMod](#) to the [dependency model](#) ([DepMod](#)), i.e., a LEFT-branching analysis of  $A B C$ ,  $[[A B] C]$  indicates that  $A$  modifies  $B$ ,

### 23. Related Work on Compound Parsing

while a RIGHT-branching analysis, [A [B C]] indicates that A “modifies something denoted primarily by” C. His work was inspired by Hindle and Rooth (1993), who performed PP attachment disambiguation based on statistical corpus evidence, Resnik and Hearst (1993), who used **Conceptual Association** (CA) for structural disambiguation, i.e., associations between semantic concepts rather than concrete instances, and Lauer and Dras (1994), who developed a probabilistic model for syntactically analysing such **compounds**. Lauer (1994) extracted 35,909 out-of-context noun pairs from Grolier’s multimedia online encyclopedia, serving as training data. For using semantic concepts, all nouns are mapped on a list of categories in Roget’s Thesaurus. Noun pairs in which there is a noun without evidence in Roget’s Thesaurus are removed from the training set, leading to 24,285 training noun pairs. For measuring the CA between two thesaurus categories  $t_1$  and  $t_2$ , Lauer (1994) defined the MI-like measures given in Formula 23.7.

$$\begin{aligned}
 \text{AMBIG}(w) &= \text{number of thesaurus categories of } w \\
 \text{COUNT}(w_1, w_2) &= \text{number of occurrences of } w_1, w_2 \text{ in the training data} \\
 \text{FREQ}(t_1, t_2) &= \sum_{w_1 \in t_1} \sum_{w_2 \in t_2} \frac{\text{COUNT}(w_1, w_2)}{\text{AMBIG}(w_1) \cdot \text{AMBIG}(w_2)} \\
 \text{CA}(t_1, t_2) &= \frac{\text{FREQ}(t_1, t_2)}{\sum_{v_i} \text{FREQ}(t_1, i) \cdot \sum_{v_i} \text{FREQ}(i, t_2)} \tag{23.7}
 \end{aligned}$$

For parsing a 3NC, A B C, Lauer (1994) first determined the thesaurus categories  $S_1$  (for  $w_1$ ) and  $T_i$  (for  $w_2$  or  $w_3$ ) such that  $\text{CA}(S_1, T_i)$  has a maximum value. If  $\text{CA}(S_1, T_3) > \text{CA}(S_1, T_2)$ , A B C is RIGHT-branching, otherwise LEFT-branching.

LEFT	RIGHT	SEMIND	ERROR	Total
163	81	35	29	308
52.9%	26.3%	11.4%	9.4%	100%
163	81	35	####	279
58.4%	29.0%	12.5%	####	100%
163	81	#####		244
66.8%	33.2%	#####		100%

Table 23.1.: Structure class distribution in Lauer (1994)

In an experiment, Lauer (1994) extracted 308 out-of-context 3NC instances from Grolier’s encyclopedia with evidence in Roget’s Thesaurus, and manually labelled them

### 23. Related Work on Compound Parsing

with one of the four categories: LEFT, RIGHT, SEMIND and ERROR. The distribution of these four categories as well as for the three structure classes (LEFT, RIGHT, SEMIND) and the two determinate classes (LEFT, RIGHT) is given in Table 23.1.

Excluding extraction errors, Lauer (1994) observed 12.5% [semantically indeterminate 3NCs](#) in their dataset. For the determinate structure classes (LEFT, RIGHT), he observed that about two-thirds are LEFT-branching (66.8%). This observation about the [LEFT class baseline](#) is in line with subsequent work on [parsing English 3NCs](#).

**Lauer (1995a)** adopted the [DepMod](#) proposed by Lauer (1994). For sampling training data (i.e., pairs of nouns), Lauer (1995a) followed two strategies: (1) how often does a pair of nouns occur in isolation:  $\text{freq}(w_1 n_1 n_2 w_4)$  where  $n_1$  and  $n_2$  are nouns and  $w_1$  and  $w_4$  are not nouns (Pustejovsky et al., 1993), and (2) how often do two nouns co-occur separated by a sequence of  $i$  words:  $\text{freq}(n_1 w_1 \dots w_i n_2)$ . Following Lauer and Dras (1994), Lauer (1995a) defines a parameter representing the degree of acceptability as given in Formula 23.8.

$$Pr(t_1 \rightarrow t_2) = \frac{1}{\mu} \sum_{\substack{w_1 \in t_1 \\ w_2 \in t_2}} \frac{\text{COUNT}(w_1, w_2)}{\text{AMBIG}(w_1) \cdot \text{AMBIG}(w_2)} \quad (23.8)$$

$$\text{where } \mu = \sum_{\substack{w_1 \in N \\ w_2 \in t_2}} \frac{\text{COUNT}(w_1, w_2)}{\text{AMBIG}(w_1) \cdot \text{AMBIG}(w_2)}$$

For deciding whether a [3NC](#) is LEFT- or RIGHT-branching, Lauer (1995a) used the ratio of LEFT- to RIGHT-branching probability, as given in Formula 23.9 for the [AdjMod](#) ( $R_{\text{AdjMod}}$ ) and the [DepMod](#) ( $R_{\text{DepMod}}$ ). If this ratio is greater than 1, the method votes for a LEFT-branching structure, and if it is less than 1, the vote is RIGHT. For the unlikely case of a ration equal to 1, the [LEFT class baseline](#) is used.

$$R_{\text{AdjMod}} = \frac{\sum_{t_i \in \text{cats}(w_i)} Pr(t_1 \rightarrow t_2)}{\sum_{t_i \in \text{cats}(w_i)} Pr(t_2 \rightarrow t_3)}$$

$$R_{\text{DepMod}} = \frac{\sum_{t_i \in \text{cats}(w_i)} Pr(t_1 \rightarrow t_2) \cdot Pr(t_2 \rightarrow t_3)}{\sum_{t_i \in \text{cats}(w_i)} Pr(t_1 \rightarrow t_3) \cdot Pr(t_2 \rightarrow t_3)} \quad (23.9)$$

In their experiments' results, Lauer (1995a) used the test set developed by Lauer (1994) (as described in Table 23.1) and observed that the sampling strategy using a 'windowed co-occurrence did not help' and that the [DepMod](#) outperforms the [AdjMod](#)

in all settings, with an accuracy of 81% (outperforming the [LEFT class baseline](#) of 66.8%).

**Lapata and Keller (2004)** demonstrated the utility of web counts as approximation for bigram corpus frequencies for six different [NLP](#) tasks, such as [Machine Translation \(MT\)](#) candidate selection, spelling correction or adjective ordering. As an analytic task, Lapata and Keller (2004) also investigated the performance of [parsing](#) English [3NCs](#). They derived web counts using hits of the Altavista<sup>1</sup> search engine, where three different types of queries were used: (1) *literal queries* with a quoted *Ngram*, (2) *near queries* using Altavista’s NEAR operator, based on a 10-word window, and (3) *inflected queries* comprising all literal queries with morphological alternatives to a given *Ngram*. As baseline, Lapata and Keller (2004) used the same model trained on frequencies from the BNC. Lapata and Keller (2004) adopted the approach of Lauer (1995a) (based on probability ratios) but used the web counts instead of corpus frequency and Roget’s Thesaurus. In their experiments’ results, Lapata and Keller (2004) observed that the web-based [compound parser](#) was significantly better than the corpus-based counterpart. However, the best Altavista model was not significantly different from the tuned model of Lauer (1995a). **Lapata and Keller (2005)** adopted the experiments of Lapata and Keller (2004) and added the [NLP](#) tasks of article restoration and [PP](#) attachment disambiguation. Moreover Lapata and Keller (2005) tried to develop an *interpolation model* based on a combination of web counts (known to be noisy) and corpus counts (known to be sparse). The interpolation model was able to outperform the tuned model of Lauer (1995a) but as for Lapata and Keller (2004), the differences were not significant.

**Nakov and Hearst (2005)** developed an unsupervised approach to [parsing 3NCs](#). They used the number of Google search engine page hits for approximating corpus frequencies. Besides the common bigrams for the [AdjMod](#) and the [DepMod](#), Nakov and Hearst (2005) defined some *surface features* that motivated us to use [split point markers](#) in the [compound splitting](#) task presented in Part D. These surface features also slightly resemble the aligned phrases used in our [cross-lingual compound parser](#). The surface features include:

- dashes (as in *cell-cycle analysis* pointing to LEFT)
- possessive markers (as in *brain’s stem cell* pointing to RIGHT)
- capitalization (as in *Plasmodium vivax Malaria* indicating LEFT)

---

<sup>1</sup>A former search engine which has ended its service in 2013.



### 23. Related Work on Compound Parsing

- slashes for marking options (as in *leukemia/lymphoma cell* pointing to RIGHT)
- parentheses (as in *(brain) stem cell* indicating RIGHT)
- punctuation such as commas (as in *health care, provider* meaning LEFT)
- acronyms (e.g., *tumor necrosis factor (NF)* pointing to RIGHT)
- optional **closed compounding** (e.g., *healthcare reform* indicating LEFT)
- internal inflection variability (e.g., *tyrosine kinases<sub>2</sub> activation* pointing to LEFT)
- switching the **word** order (e.g., evidence for BAC as *male adult rat* being an alternative for *adult male rat* indicates RIGHT)

But there is another type of feature used by Nakov and Hearst (2005) which resembles the aligned phrases in our approach even more, the *paraphrase features*. Besides some Google wildcard queries modeling most possible paraphrases (having one or more **words** between the **constituents**), Nakov and Hearst (2005) defines three types of paraphrases: (1) prepositional phrases (as in *cells<sub>C</sub> from the brain<sub>A</sub> stem<sub>B</sub>* pointing to LEFT), (2) copula paraphrases (as in *office<sub>B</sub> building<sub>C</sub> that/which is a skyscraper<sub>A</sub>* indicating LEFT) and (3) verbal paraphrases (as in *pain<sub>C</sub> associated with arthritis<sub>A</sub> migraine<sub>B</sub>* meaning LEFT). In all of these paraphrase types, there is a complex unit of A B or B C, which we exploit directly in the **Aligned Phrase Pattern Parsing (APPP)** and indirectly using the **aligned word distance (AWD)** metric.

In their experiments, Nakov and Hearst (2005) compared three **Association Measures (AMs)** for the bigrams: plain frequency (as used by Lapata and Keller (2004)), probabilities (as used by Lauer and Dras (1994) and Lauer (1995a)) and **Chi Squared ( $\chi^2$ )**. They observed that  $\chi^2$  is the best-working **AM** for **parsing 3NCs**, which is in line with Yang and Pedersen (1997), who showed that  $\chi^2$  outperforms **Mutual Information (MI)** as **AM**. While Nakov and Hearst (2005) showed that the **DepMod** clearly outperforms the **AdjMod** in the test set provided by Lauer (1994), in their own **3NC** test set compiled from the MEDLINE abstracts, the performance of **AdjMod** and **DepMod** show comparable performance. Nakov and Hearst (2005) observed that the surface features are good at predicting LEFT-branching **compounds**, “but unreliable for RIGHT-bracketed examples”. Their knowledge-rich approach including surface and paraphrase features clearly outperforms previous state-of-the-art approaches to **compound parsing**. Their method is “more robust than Lauer (1995a) and more accurate than Lapata and Keller (2004)” (Nakov and Hearst, 2005).

In the following experiments on [compound parsing](#) (Chapter 25) we aim to compare the knowledge-lean state-of-the-art approach based on the  $\chi^2$  measure with our knowledge-lean [parsing](#) methods, which allow for a fair comparison without relying on manual resources such as hand-crafted paraphrase or surface features. Moreover, we do not aim to compare different [compound parsers](#) but different measures for [parsing](#), i.e., the [AWD](#) metric against the strongest monolingual [AM](#) discussed for [compound parsing](#):  $\chi^2$ . Additional monolingual features are plausible for both  $\chi^2$  and [AWD](#).

[Girju et al. \(2005\)](#) presented several supervised and unsupervised models for [parsing](#) and the semantic interpretation of [2NCs](#) and [3NCs](#). As an unsupervised [compound parser](#), Girju et al. (2005) adopted the web-based approach of Lapata and Keller (2004). As a supervised [compound parser](#), a C5.0 decision tree model is employed with 15 ‘linguistic features’ based on WordNet senses, five for each noun [constituent](#):

1. **WordNet derivationally related form:** this feature specifies if the [constituent](#)’s sense is derived from a verb (e.g., *coffee maker industry*)
2. **WordNet top semantic class:** this feature provides the top category of the [constituent](#)’s sense (e.g., *coffee maker industry*  $\rightarrow$  *group/grouping*)
3. **WordNet second top semantic class:** this feature provides the second top category of the [constituent](#)’s sense (e.g., *coffee maker industry*  $\rightarrow$  *social\_group*)
4. **WordNet third top semantic class:** this feature provides the third top category of the [constituent](#)’s sense (e.g., *coffee maker industry*  $\rightarrow$  *organization*)
5. **Nominalization:** this feature indicates whether the [constituent](#) is a nominalization. Therefor, Girju et al. (2005) used the NomLex dictionary (Macleod et al., 1998) and the `event/action` categories in WordNet

Girju et al. (2005) sampled [3NCs](#) *in the context* of the WSJ articles from TREC-9 and of eXtended WordNet glosses (XWN 2.0), where [compounds](#) were semi-automatically annotated. The supervised approach of Girju et al. (2005) achieves a [parsing](#) accuracy of 83.1% and clearly outperforms the unsupervised method of Lapata and Keller (2004) (both as [AdjMod](#) (73.5%) and as [DepMod](#) (77.4%).

Similar to Girju et al. (2005), [Kim and Baldwin \(2013\)](#) presented a study on the interpretation and [parsing](#) of bipartite and tripartite [noun compounds](#) using lexical semantics. Kim and Baldwin (2013) investigated whether “[NC](#) interpretation predictions can enhance [NC bracketing](#)”. This [parser](#) is based on a method that determines the

semantic relation (SemRel) for 3NCs and 2NCs, also described in Kim and Baldwin (2013).

In the first step, the two “outermost” 2NCs are extracted from a given 3NC  $A\ B\ C$ , i.e.,  $A\ C$  (reflecting the heads for the SemRel given a RIGHT-branched interpretation) and  $B\ C$  (reflecting the heads for the SemRel given a LEFT-branched interpretation). In the second step, the SemRels for  $A\ C$ ,  $B\ C$  and  $A\ B\ C$  are classified. If the SemRel( $A\ B\ C$ ) is equal to that of SemRel( $A\ C$ ) and not to that of SemRel( $B\ C$ ), the parser votes for a RIGHT-branched structure, and vice versa for LEFT. Kim and Baldwin (2013) exemplified their algorithm with the 3NC, *physics winter school*, having the two outermost 2NCs *physics school* (i.e.,  $A\ C$ ) and *winter school* (i.e.,  $B\ C$ ). Using the semantic interpreter proposed by Kim and Baldwin (2013), the 3NC, *physics winter school*, as well as the 2NC, *physics school*, get the SemRel TOPIC, whereas *winter school* gets the SemRel TIME. Since  $\text{SemRel}(\textit{physics winter school}) = \text{SemRel}(\textit{physics school}) \neq \text{SemRel}(\textit{winter school})$ , the parser votes for a RIGHT-branched bracketing: *physics [winter school]*. Since relying on WordNet (in order to measure semantic similarity) suffers from coverage, Kim and Baldwin (2013) combined their method with the probabilistic model of Lauer (1995a) and with the state-of-the-art method of Nakov and Hearst (2005), i.e., they back off to their semantic-based parser if they do not have any instances of both word pairs. Kim and Baldwin (2013) observed that the combined models outperform the isolated models both in coverage and accuracy. The best model is the combination with Nakov and Hearst (2005) and achieves roughly full coverage.

Similar to Lapata and Keller (2004), Bergsma et al. (2010) assessed the benefit of the addition of web-scale Ngram features to various supervised NLP classifiers and its applicability on new domains. Besides adjective ordering, spelling correction and verb PoS disambiguation, the inspected NLP tasks included the parsing of 3NCs.

As source for the Ngram features, Bergsma et al. (2010) used Google V2 (Lin et al., 2010), and as *in-domain* training, development and test set, Bergsma et al. (2010) extracted 2150 samples of 3NCs from the dataset of Vadas and Curran (2007a). As *out-of-domain* test set, Bergsma et al. (2010) used the test set of Lauer (1994) and of Nakov (2007). As supervised bracketing classifier, Bergsma et al. (2010) used a linear Support Vector Machine (SVM), trained with lexical features and optionally Ngram features. As lexical features, the nouns and their position, all three noun pairs, the entire noun triple as well as a capitalization pattern of the noun sequence are used. As Ngram features, the log corpus frequency, based on Google V2, of all constituent subsets were used. Moreover, the frequency of closed compounds derived from concatenating

adjacent **words** of the underlying **compound** (Nakov and Hearst, 2005) were included.

Bergsma et al. (2010) compared the supervised system against an unsupervised method relying on the **PMI** of **word** pairs according to the **DepMod**, and against the **LEFT class baseline**. As result, Bergsma et al. (2010) observed that the best method in comparison for most test sets is the **SVM** model based on a combination of *N*gram features and lexical features, which significantly outperforms the usage of only lexical features, in particular when switching to the *out-of-domain* test sets.

Besides English **open compounds**, there is also previous work addressing the **parsing** of **noun compounds** and noun sequences in Indian languages such as Sanskrit, Hindi and Marathi. Previous work on **parsing** Indian noun sequences include **Kulkarni and Kumar (2011)**, **Kulkarni et al. (2012)**, **Kavuluru and Harris (2012)**, **Batra et al. (2014)** and **Batra and Paul (2015)**.

## 23.4. Parsing of Base NPs

**Barker (1998)** presented a semi-automatic method for **bracketing** a list of **modifiers** (“noun premodifiers”) for a given **base NP** of any **size** (in terms of **atomic constituents**). The algorithm uses a sliding window of three **constituents** (either **atomic** or complex) and thus reduces the task of **parsing** *k*-partite **NPs** to the task of **parsing** three-word **NPs**.

1. As starting point, the sliding window covers the last three **constituents** (e.g., X Y Z):

$$\dots \quad V \quad W \quad \overline{X \quad Y \quad Z}$$

- 2a. If there is evidence for a **RIGHT**-branched **XYZ**, Y and Z are merged and the window introduces the **constituent** left to X:

$$\dots \quad V \quad \overline{W \quad X \quad YZ}$$

- 2b. If there is evidence for a **LEFT**-branched **XYZ**, the sliding window is moved one position to the left to cover the **constituents** W, X and Y:

$$\dots \quad \overline{V \quad W \quad X \quad Y \quad Z}$$

The [constituents](#) X and Y are not automatically merged, because they do not necessarily be direct siblings (e.g., in the case that W and X form a [constituent](#)).

- When the sliding window reaches the first [constituent](#), evidence for a LEFT-branching structure leads to the immediate fusion of the two leftmost [constituents](#) and the sliding window introduces the [constituent](#) to the right:

$$\overline{UV \quad W \quad X \quad Y \quad \dots}$$

Barker (1998) described the following rules for LEFT- and RIGHT-branching evidence of a three-[constituent](#) sequence (covered by the sliding window):

- noun adj noun sequences are usually RIGHT-branching, because adjectives are mostly prenominal and only very infrequently postnominal
- the three-[constituent](#) sequences X-Y-Z are reduced to their heads  $X_h$ - $Y_h$ - $Z_h$ . If  $freq(X_h, Z_h) > \theta \cdot freq(X_h, Y_h)$  there is evidence for a RIGHT-branching window content, where *freq* refers to the observed frequencies of the [parser](#) for previous analyses and  $\theta$  is a predefined threshold
- if  $freq(X_h, Y_h) > \theta \cdot freq(X_h, Z_h)$  it is LEFT-branched.
- if there is no evidence for a LEFT- or RIGHT-branching structure, Barker (1998) proposes two options:
  - in a semi-automatic way, the user is consulted
  - in a fully automatic way, the system decides for the [LEFT class baseline](#), which is only reliable for sequences of three nouns

In this algorithm, Barker (1998) compared XY against XZ and thus followed the principle of the [DepMod](#).

As an example for illustrating the process of [NP parsing](#), Barker (1998) used the phrase *wooden French onion soup bowl handle*, i.e., the wooden handle of a bowl for a French soup made with onions. Barker (1998) assumed that the [parser](#) already determined *soup bowl* and *wooden [pot handle]*.

In the initial configuration, the sliding window covers *soup bowl handle*:

### 23. Related Work on Compound Parsing

*wooden French onion soup bowl handle*

Knowing the [2NC](#) *soup bowl*, there is a LEFT-branching evidence and thus the window moves left:

*wooden French onion soup bowl handle*

The system cannot get an automatically derived evidence for a LEFT- or RIGHT-branching structure of *onion soup bowl*. Falling back to the [LEFT class baseline](#) or consulting the user would lead to a LEFT-branching decision and thus the window moves one step left:

*wooden French onion soup bowl handle*

Since there is no evidence for a LEFT- or RIGHT-branching *French onion soup*, the user consultation results in a RIGHT-branching decision. Thus, *onion* and *soup* get merged and the window expands one [constituent](#) to the left:

*wooden French [onion soup] bowl handle*

Since *French* being an adjective, the system votes for a RIGHT-branching *wooden French [onion soup]*. As the leftmost [constituent](#) is already covered by the window, it expands one [constituent](#) to the right:

*wooden [French [onion soup]] bowl handle*

In the next step, the [atomic head](#) of *French onion soup* is considered. Since neither *wooden soup* nor *wooden bowl* is known, the system consults the user, leading to a RIGHT-branching decision, i.e., *French onion soup* and *bowl* are merged. Again, the window is expanded to the right:

*wooden [[French [onion soup]] bowl] handle*

In the final step, there is only a three-[constituent parsing](#) task left with the [constituents](#) *wooden*, *[[French [onion soup]] bowl]* and *handle*. Considering only the heads, the system inspects the [word](#) pairs (*wooden*, *bowl*) and (*wooden*, *handle*). As mentioned above, the

### 23. Related Work on Compound Parsing

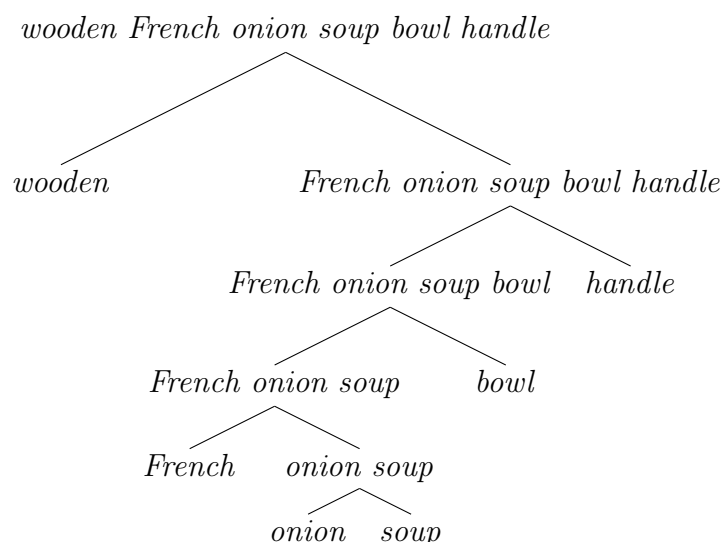


Figure 23.3.: Parse tree for *wooden French onion soup bowl handle*

system already knows the NP *wooden handle* and thus the final bracketing is RIGHT, leading to the parse tree as given in Figure 23.3.

While Barker (1998) designed the method for English NPs, it should be applicable on other target languages without much adaptation (e.g., adjusting the likelihood of prenominal and postnominal adjectival modifiers). The parser analyzes NPs out-of-context and thus neglects structural ambiguity. Since the system learns bracketing from user consultation, it can be considered as a semi-supervised parsing approach.

In some experiments, Barker (1998) observed that his semi-automatic method was able to parse 62-65% of all NP samples correctly without human support. Another result was that “as more compounds are bracketed, the number of bracketing decisions required of the user decreases” (Barker, 1998).

Pitler et al. (2010) criticized the approach of Barker (1998), because three-word sequences cannot always be parsed in isolation, in particular not for coordinations as in *[[soap opera] stars] and [television producers]* and *[[movie and television] producers]*, where the last three words *and television producers* are structured differently.

Vadas and Curran (2007a) manually added a gold-standard bracketing of base NPs to the Penn Treebank (PTB) (Marcus et al., 1993). The original PTB only provides flat structures for base NPs due to the annotation effort. For example, the 3NC *Air Force contract* is originally represented as:

```
(NP
(NNP Air) (NNP Force) (NN contract))
```

)

In contrast, the novel annotation scheme of Vadas and Curran (2007a) provides the information that *Air Force contract* is LEFT-branching and *Air Force* is a nominal **modifier** (NML) of *contract*:

```
(NP
(NML (NNP Air) (NNP Force))
(NN contract)
)
```

In their annotation scheme, Vadas and Curran (2007a) kept RIGHT-branching constructions untouched. Two labels were introduced for marking a nominal **modifier** (NML) or an adjectival **modifier** (JJP) in a LEFT-branching structure.

In later revisions of the annotation guidelines (Vadas, 2009), additional labels for FLAT (e.g., proper names like *John A. Smith*) or **semantically indeterminate** constructions were added.

**Vadas and Curran (2008)** improved the **NP** structure annotations in the CCGbank (Hockenmaier and Steedman, 2007) using an automatic conversion process applied to the data of Vadas and Curran (2007a), leading to a more accurate representation of the **NPs** in CCGbank and thereby to a higher **parsing** performance.

**Vadas and Curran (2007b)** developed several large-scale models using the **PTB** (annotated with **NP** structure by Vadas and Curran (2007a)) for **parsing base NPs**.

They extracted 5582 annotated three-word **NPs** from the **PTB**, a set of samples which is an order of magnitude larger than those used in previous work, allowing for a sophisticated machine learning model rather than using an unsupervised approach. Vadas and Curran (2007b) also created a set of 36,584 more complex **NPs** (comprising three or more **words**) from the **PTB**, which is two orders of magnitude larger than the datasets used in previous work.

Vadas and Curran (2007b) presented an unsupervised method for **parsing** the three-word **NPs**. For counting bigrams, they used three sources: hit counts from the web search engines Google and MSN, and frequencies in the Google Web 1T corpus (Brants and Franz, 2006). Plain word pairs (and some variations according to Nakov and Hearst (2005)) were compared according both to the **adjacency model** (**AdjMod**) and the **dependency model** (**DepMod**). As **Association Measures** (**AMs**) Vadas and Curran (2007b) experimented with raw frequency, bigram probability ( $P(w_i, w_j | w_j)$ ) and  $\chi^2$  measure.



The best AM was  $\chi^2$ , being in line with the observations made by Nakov and Hearst (2005). For the three-word NP dataset, the AdjMod outperforms the DepMod. We made similar observations for the 3NC dataset from the ENCD.

The best result of the unsupervised method achieves 83.61% on the dataset of Lauer (1995a). While the performance of the system developed by Nakov and Hearst (2005) is higher (89.3%), the unsupervised approach of Vadas and Curran (2007b) does not rely on knowledge such as paraphrases.

The first supervised model developed by Vadas and Curran (2007b) is based on a MegaM Maximum Entropy (ME) classifier (Daumé III, 2004) trained on the PTB. As features, they used all counts, probabilities and metrics from the unsupervised method (for both AdjMod and DepMod; i.e., they combine them to an Adjacency-Dependency Model (AdjDepMod)). This model outperforms the unsupervised model by 6.45% in F<sub>1</sub>-Score. The advantage of the supervised model is due to its “ability to weight the individual contributions of all of the unsupervised counts from Google and the Web 1T corpus” (Vadas and Curran, 2007b).

As a second supervised model, Vadas and Curran (2007b) added lexical features for all bigrams and trigrams in the NP along with their position. In addition, they used contextual features: a bag-of-words feature for the surrounding sentence and features for a two-word window around the NP. For each Ngram and context window, a generalized version is added by replacing each word with the corresponding PoS or NER tag. Finally, semantic features derived from WordNet (Fellbaum, 1998) were added, such as synsets for each sense of all words (and their hypernyms) in the NP. Vadas and Curran (2007b) observed that the lexical and NER features “are most important but all make a positive contribution”. The best performance is achieved when using all features (F<sub>1</sub>-Score of 93.01%, outperforming the unsupervised model by 8.87%).

For processing more complex NPs, Vadas and Curran (2007b) implemented the algorithm of Barker (1998). They presented several models for determining whether the three-word window in Barker’s algorithm is LEFT- or RIGHT-branching. As unsupervised model, they used the  $\chi^2$  metric in the AdjMod and DepMod. As supervised model, Vadas and Curran (2007b) applied the supervised model developed for three-word NPs on Barker’s window. In an experiment, Vadas and Curran (2007b) observed that all supervised models clearly outperformed all unsupervised models by far. For example, using all features with 500 iterations in MegaM achieved a matched bracket F<sub>1</sub>-Score of 91.44%, outperforming  $\chi^2$ -DepMod (32.40%) by 59.04%.

Pitler et al. (2010) presented an automatic supervised parser for base NPs of any

length including coordinations. In particular **base NPs** with coordinations pose a big challenge, e.g., in *French television and movie producers*, the center **words** *television and movie* have to be grouped. Pitler et al. (2010) developed a supervised efficient linear **Support Vector Machine (SVM)**<sup>2</sup> classifier for estimating the probability of a **word** sequence being a **constituent** within the context of the entire **NP**. These probabilities are inserted into a chart. For scoring a possible **parse tree**, Pitler et al. (2010) multiplied the probabilities of all **atomic** and complex **constituents** in the tree. For determining the most probable **parse tree**, the CYK algorithm is used (Pitler et al., 2010). An advantage over most previous work is that this chart allows for a global perspective on the full **base NP**. The chart of **bracketing** scores can be integrated easily into a downstream full sentence parser or applied directly on a chart parser (Pitler et al., 2010).

As features for their classifier, Pitler et al. (2010) used the position of the **bracketing** relative to the full **NP**. **PMI** features are used for all **word** pairs in the **NP**, derived from the web-scale *N*gram corpus Google V2 (Lin et al., 2010). For some particular **words** (e.g., *Inc.*) a binary lexical feature indicates the position (e.g., *Inc.* is usually outside brackets). Another group of features concern the shape of the bracketed **word** group: indicating capitalized letters and hyphenated **words**, it is possible to get **NER** information without the need for **NER** training data (as has been used by Vadas and Curran (2007b)). Unfortunately, Pitler et al. (2010) did not care about structural ambiguity (e.g., by taking into account contextual features for disambiguation).

As training and test data, Pitler et al. (2010) used the **NPs** in the **PTB** annotated by Vadas and Curran (2007a). For each **NP**, positive or negative samples of (complex) **constituents** are generated with their feature values. The method of Pitler et al. (2010), which achieves an **NP parsing** accuracy of 95.4%, outperforms a baseline which always predicts a RIGHT-branching **parse tree** (72.6%) by 22.8%. “The most comparable result is by Vadas and Curran (2007b), who achieved 93.0% accuracy on a different set of **PTB noun phrases**, but their classifier used features based on gold-standard part-of-speech and named-entity information” (Pitler et al., 2010).

**Lazaridou et al. (2013)** addressed the **parsing** of English three-**word NPs** (where the first **word** can be either an adjective or a noun and the other two **words** are nouns) using a measure of semantic plausibility as derived from **Distributional Semantics (DS)**. For example, knowing that *home run* is semantically more plausible than *miracle home* leads to a RIGHT-branched *miracle [home run]*. The vectors of each **constituent** are combined (as *basic composition* between **atomic words** or as *recursive composition* including an

---

<sup>2</sup>[www.csie.ntu.edu.tw/~cjlin/liblinear/](http://www.csie.ntu.edu.tw/~cjlin/liblinear/)

already composed *constituent*) using a composition function. Lazaridou et al. (2013) used an *SVM* with a Radial Basis Function kernel and as features, they used the semantic plausibility derived from basic and from the recursive composition, and the *PMI* values for the *word* pairs (according to the *AdjMod*). Lazaridou et al. (2013) showed that their approach based on semantic plausibility and *PMI* values, which achieves an accuracy of 85.6%, significantly outperforms a statistical baseline relying only on *PMI* (81.2%) or relying only on semantic plausibility (78.7%). Moreover, Lazaridou et al. (2013) compared their method against a *RIGHT*-branching baseline (65.6%) and a *PoS*-based baseline (77.3%), which predicts *LEFT* for *noun-noun-noun* sequences (cf. *LEFT class baseline* for *3NCs*) and *RIGHT* for *adjective-noun-noun* sequences.

**Ménard and Barrière (2014)** presented an unsupervised *parser* for out-of-context *NPs*<sup>3</sup> based on an association model. Instead of focusing on a three-*word* window (as suggested by Barker (1998)), Ménard and Barrière (2014) compared all possible *word* pairs within the *NP*, allowing for any long-range dependencies. In our *cross-lingual compound parsing* methods, we also take into account long-range dependencies as much as the *word* order of *constituent* equivalents can change across languages (as will be exemplified in Section 25.3.3 for *church<sub>A</sub> development<sub>B</sub> aid<sub>C</sub> projects<sub>D</sub>* being aligned to the Italian *progetti<sub>D</sub> ecclesiastici<sub>A</sub> di aiuti<sub>C</sub> allo sviluppo<sub>B</sub>*, where there is a long-range dependency relation between *church* and *projects*).

Ménard and Barrière (2014) compared the usage of three different resources. The first two are based on *Ngram* frequency: the English Google Web *Ngrams* (Lin et al., 2010), the English (non-fictional) Google Books *Ngrams* (Michel et al., 2010), and the third is the open linked data DBpedia V3.9 (Hellmann et al., 2009), which is based on automatically parsed Wikipedia infoboxes.

As frequency-based *AMs* relying on the *Ngram* corpora, Ménard and Barrière (2014) compared the  $\chi^2$ , the *PMI* and the Dice measure. As *AM* based on DBpedia, Ménard and Barrière (2014) used the number of valid DBpedia paths (as defined in Ménard and Barrière (2014, Sec. 5.2)) for the entities denoted by the *constituents*.

In their algorithm, a list of all *word* pairs is generated. In a second list, dependencies between *word* pairs having the highest *AM* value are iteratively added from the *word* pair list for creating a final dependency *parse tree*. The *word* pairs to be added must comprise an unused *modifier* and must not create a crossing of any already collected *modifier/head* pairs. The algorithm stops if all but the last of the *words* in the *NP* has

<sup>3</sup>Although, Ménard and Barrière (2014) claimed to process *noun compounds* of any *size*, they extracted a gold-standard from the *NP* dataset of Vadas and Curran (2007a) without considering the property of *compoundhood*.

been used as [modifier](#) in a collected dependency.

As experimental results, Ménard and Barrière (2014) observed that their method outperforms baselines predicting always RIGHT or LEFT. Moreover, they claim to outperform the unsupervised approaches of Vadas and Curran (2007b) in exact match accuracy.

**Barrière and Ménard (2014)** adopted the [NP parser](#) developed by Ménard and Barrière (2014) but used an association model relying on information provided with Wikipedia.

As a contribution, Barrière and Ménard (2014) differentiated several subtypes of [word](#) association:

**Basic dependency association:** the association based on co-occurrence in a corpus.

Barrière and Ménard (2014) used [PMI](#) and Dice as [AMs](#) on the full English Wikipedia corpus.

**Relational association:** the association based on indications of a possible [semantic relation](#) between two [words](#). Barrière and Ménard (2014) used a simple pattern composed of the two [words](#)  $w_1$  and  $w_2$  connected via a preposition [prep](#):  $w_1$  [prep](#)  $w_2$ , where [prep](#)  $\in \{about, at, by, for, from, in, of, on, to, with\}$  and counted the Wikipedia corpus frequency of these patterns.

**Coordinate association:** the association based on a relation between the [modifiers](#) in a coordinate compound (cf. Section 3.7.1). Evidence of such a coordinate relation would decrease the dependency score for the [parsing](#) task. Similar as for the *relational association*, Barrière and Ménard (2014) used a simple pattern composed of the two [words](#)  $w_1$  and  $w_2$  connected via a conjunction [conj](#):  $w_1$  [conj](#)  $w_2$ , where [conj](#)  $\in \{or, and, nor\}$ . The higher the corpus frequency of matched pattern instances, the lower the resulting dependency score between the underlying [words](#).

**Lexical association:** the association based on the probability that a subexpression forms a lexical unit. Barrière and Ménard (2014) described two approaches for measuring the *lexical association*. Firstly, the *statistical approximation*, where the frequency of the patterns ‘DET  $w_1$   $w_2$ ’, where DET  $\in \{a, an, the\}$  (i.e., a determiner introducing the underlying [word](#) pair), ‘ $w_1$  plural( $w_2$ )’, where plural(...) denotes the plural form (i.e., the underlying [word](#) pair with a pluralized second [word](#)) and ‘DET  $w_1$  plural( $w_2$ )’ are used for measuring lexical association. Secondly, the *presence in Wikipedia*, where Barrière and Ménard (2014) collected all Wikipedia page titles and checked for evidence of any subexpressions as Wikipedia page title.

In an experiment, Barrière and Ménard (2014) compared the contribution of the different subtypes of [word](#) association with the baseline of using only the basic dependency association. For all association types (i.e., relational, coordinate and lexical), there is a small or only marginal improvement over the baseline, “but it does not give a clear view of whether” their “corpus-based approximations” on the [word](#) associations “are correct or not” (Barrière and Ménard, 2014).

## 23.5. Cross-lingual Disambiguation of other Structures

To the best of our knowledge, we are the first who used [cross-lingual](#) evidence for [parsing noun compounds](#). All previous [compound parsing](#) approaches, described above, use monolingual information (usually corpus frequency). However, [cross-lingual](#) information has been used previously for [parsing](#) other types of expressions, i.e., disambiguating their internal structure.

**Yarowsky and Ngai (2001)** projected [PoS](#) tags and the chunks of [base NPs](#) from English to Chinese and French.

**Schwartz et al. (2003)** developed an unsupervised approach to resolving the [PP](#) attachment ambiguity in English using the alignments to Japanese derived from a [parallel corpus](#).

**Smith and Smith (2004)** combined statistical dependency parsers with [Probabilistic Context-Free Grammars](#) (PCFGs) and [word-to-word](#) translation models into a bilingual parser that is capable of jointly determining the best sentence structure for English and Korean.

**Hwa et al. (2005)** addressed the lack of syntactic annotations (necessary for the automatic training of a statistical parser) for languages other than English and proposed to project English [parse trees](#) to other languages for bootstrapping statistical non-English parsers. In two studies, Hwa et al. (2005) induced a Spanish and a Chinese parser.

Similar to Schwartz et al. (2003), **Fossum and Knight (2008)** resolved English [PP](#) attachment ambiguity using the support of Chinese, in which there is no such an ambiguity, leading to an accuracy of 86.3% in the [PP](#) attachment disambiguation, outperforming the Collins parser baseline.

**Burkett and Klein (2008)** developed a [ME](#) bitext parsing model based on source and target [parse trees](#) and a node-to-node alignment between them. Burkett and Klein

(2008) applied their model on the English-Chinese language pair and substantially outperformed the monolingual parsers for both sides.

**Snyder et al. (2009)** presented an unsupervised sentence parsing method based on bilingual [parse tree](#) alignments derived from a [parallel corpus](#). Therefore, they proposed a Bayesian model “which seeks to explain the observed [parallel data](#) through a combination of bilingual and monolingual parameters”. In an experiment for the language pairs Korean-English, Urdu-English and Chinese-English, Snyder et al. (2009) observed substantial performance gains over a monolingual parsing baseline. An alternative way for [multilingual](#) grammar induction is addressed by **Berg-Kirkpatrick and Klein (2010)**, who did not exploit the support of [parallel corpora](#). Instead, they used a phylogeny-structured model of parameter drift. Berg-Kirkpatrick and Klein (2010) achieved substantially better results than the independent learning in eight languages, including Dutch, Swedish, Spanish, Portuguese, Slovene and Chinese.

**Iwata et al. (2010)** proposed a way for extracting a [cross-lingually](#) valid grammar from non-[parallel multilingual](#) corpora. As monolingual grammar model, Iwata et al. (2010) used [PCFGs](#). These [PCFGs](#) are assumed to be derived from a general model which is common across languages. In their experiments, Iwata et al. (2010) demonstrated the feasibility of their approach for eleven western European languages.

**Schwarck et al. (2010)** developed a [cross-lingual](#) method for differentiating subjects from objects in German sentences using an English-German bitext. In their algorithm, Schwarck et al. (2010) exploited the English [word](#) order (commonly subject-verb-object (SVO)) which allows for easily project the English subject to German using statistical [word](#) alignment. For example, the German sentence *Die Maus<sub>Subj|Obj</sub> jagt die Katze<sub>Subj|Obj</sub>* is ambiguous with respect to two readings: (1) ‘the cat is chasing the mouse’ and (2) ‘the mouse is chasing the cat’. An alignment to the first reading implies that *die Katze* is the subject and *die Maus* is the object, and for an alignment to the second reading vice versa.

**Bergsma et al. (2011)** addressed the task of coordination disambiguation, i.e., whether there is an ellipsis in a binary coordination of the form  $w_1$  and  $w_2$   $h$ , i.e., a sequence of a single word, followed by a conjunction and a [2NC](#). For example, while *rocket<sub>w\_1</sub> and mortar<sub>w\_2</sub> attacks<sub>h</sub>* includes an ellipsis for *rocket<sub>w\_1</sub> attacks<sub>h</sub>*, the coordination *asbestos<sub>w\_1</sub> and polyvinyl<sub>w\_2</sub> chloride<sub>h</sub>* does **not** imply *asbestos<sub>w\_1</sub> chloride<sub>h</sub>*. Besides using monolingual association models, Bergsma et al. (2011) showed that [cross-lingual](#) evidence in terms of surface variation is a promising feature for resolving coordination ambiguity. For example, the elliptical expression *dairy and meat production* can be resolved using

### 23. Related Work on Compound Parsing

the Finnish translation *maidon- ja lihantuotantoon* ‘milk- and {meat production}’, where the hyphen indicates the ellipsis. Bergsma et al. (2011) made use of small amounts of annotated data on the target side and complement this with bilingual features from unlabeled bitext in a co-trained classifier.

### 23. *Related Work on Compound Parsing*



# 24. Pilot Study using Aligned Phrase Patterns

In this chapter, we present a pilot study on [cross-lingual compound parsing](#), which was published in Ziering and Van der Plas (2014).

## 24.1. Aligned Phrase Patterns

In this pilot study, we developed a [token-](#) and pattern-based [parsing](#) approach which uses [universal surface patterns](#) (USPs) that model [phrasal equivalents](#), that are [cross-lingually](#) aligned to the [target compound](#), so-called [Aligned Phrase Patterns](#) (APPs). USPs have already been used for [Cross-lingual Compound Inspection](#) (XCI) in Section 10.2. The nature of USPs and the transformation from [PoS patterns](#) to USPs are discussed in Appendix A.

### 24.1.1. Function of Aligned Phrase Patterns

In the style of the paraphrase features proposed by Nakov and Hearst (2005), the intention of pattern-based [parsing](#) is that the APPs reveal an unbalanced strength of [semantic association](#) between the [target constituents](#). For example, the USP SN FC CN (i.e., a simplex noun followed by a functional context and a complex noun (i.e., a [nominal compound](#))) points to a LEFT-branched TC (e.g., the 3NC *human rights violation* being aligned to the German *Verletzung<sub>SN</sub> von<sub>FC</sub> Menschenrechten<sub>CN</sub>* (lit: ‘violation of {human rights}’). Practically, we aim to find a **complex unit** (e.g., a [closed compound](#)) in an aligned paraphrasing USP that corresponds to a complex [head](#) (i.e., a RIGHT-branched structure) or a complex [modifier](#) (i.e., a LEFT-branched structure) in the target TC.

### 24.1.2. Manual Definition of Aligned Phrase Patterns

Inspired by the [cross-lingual](#) observations about phrasal translations discussed in Section 5.2, the [Cross-lingual Compound Inspection](#) in Section 10.2 and the discussion about the most frequent paraphrase structures in our [Europarl Nominal Compound Database \(ENCD\)](#) in Section 12.1.3, we define a set of **six APPs**. The most important [USP](#) tags used in the following patterns are described in Table 24.1.

USP tag	Description
CN	complex noun
ADJ	adjective
SN	simplex noun
FC	sequence of <a href="#">function words</a>

Table 24.1.: Description of USP tags

The ten most frequent [3NC](#) paraphrase [USPs](#) for various [support languages](#) have been given in Table 12.5, repeated in Table 24.2.

German	Swedish	French	Italian
CN	CN	SN FC SN FC SN	SN FC SN FC SN
<b>ADJ CN</b>	<b>ADJ CN</b>	SN FC SN ADJ	SN FC SN ADJ
<b>SN FC CN</b>	SN	SN FC SN	SN ADJ
SN	ADJ SN	SN ADJ	SN FC SN
ADJ SN	SN SN	SN SN FC SN	SN SN FC SN
SN SN	<b>SN FC ADJ SN</b>	SN SN SN	<b>SN ADJ FC SN</b>
<b>SN FC ADJ SN</b>	PC CN	SN FC SN SN	SN SN SN
SN SN SN	PC SN	<b>SN ADJ FC SN</b>	SN FC SN SN
<b>CN FC SN</b>	SN SN SN	SN	SN
CN FC CN	SN VB SN	SN SN ADJ	SN SN ADJ

Table 24.2.: The 10 most frequent paraphrases of English [3NCs](#) in [USP](#) format

Table 24.3 shows the six [APPs](#). The first five [APPs](#) are among the ten most frequent paraphrasing [USPs](#) in the [ENCD](#), highlighted in Table 24.2. Although the last [APP](#) in Table 24.3 is not listed in Table 24.2, we added it (sixth row), because it is a plausible [RIGHT](#)-branching counterpart (pointing to a complex [head](#)) of the [LEFT](#)-branching [APP](#) in the fourth row (pointing to a complex [modifier](#)).

	Aligned Phrase Pattern	Assigned structure class
(1)	ADJ CN	RIGHT
(2)	CN FC SN	RIGHT
(3)	SN FC CN	LEFT
(4)	SN FC ADJ SN	LEFT
(5)	SN ADJ FC SN	RIGHT
(6)	ADJ SN FC SN	RIGHT

Table 24.3.: Six APPs and the corresponding structure

We discarded a possible seventh APP, which is the LEFT-branching counterpart of the fifth APP: SN FC SN ADJ (i.e., a simplex noun followed by a functional context and a simplex noun with a postnominal adjective). While the predominant structure of TCs aligned to this APP is indeed LEFT (i.e., the final adjective refers to the last noun), there is also a significant amount of RIGHT-branching interpretation (i.e., the final adjective refers to the preceding complex nominal: SN FC SN). Therefore, we decided to disregard this ambiguous APP in favor of precision and at the cost of coverage. The corresponding ambiguity for the prenominal adjective in the sixth APP (i.e., the initial adjective refers to the first noun or to the full complex nominal) does not have any impact on the structure class assignment (i.e., the prenominal adjective modifies the head or the entire complex nominal).

The examples in Table 24.4 illustrate instances for each of the six APPs in Table 24.3.

### 24.1.3. Structure Class Assignment

The selection of the six APPs listed in Table 24.3 was driven by the principle of a complex unit, discussed in Section 24.1.1. Each APP contains a complex unit which is separated from the rest, e.g., a closed nominal compound (CN) or a combination of single noun and pre- or postnominal adjective (ADJ SN or SN ADJ). The separator in APPs including closed compounds is the word boundary, whereas for complex multiword units, the functional context (FC) is used for separating the complex from the simplex unit.

The last column in Table 24.3 shows the assigned structure class (i.e., LEFT or RIGHT) of the respective APP. In order to assign this class, we first have to determine the head of the APP. For this purpose, we used a universal heuristic for each APP:

- if there is only one nominal token, it is defined as head (as in ADJ CN)







APP				Examples
ADJ	CN			German (  ): <i>staatliche</i> <sub>ADJ</sub> <i>Steueraufsichtsbehörden</i> <sub>CN</sub> 'state tax inspectorates'
CN	FC	SN		German (  ): <i>Absatzmarkt</i> <sub>CN</sub> <i>für</i> <sub>FC</sub> <i>Fahrzeuge</i> <sub>SN</sub> (lit: 'sales market for cars') 'car sales market'
SN	FC	CN		Dutch (  ): <i>methode</i> <sub>SN</sub> <i>voor</i> <sub>FC</sub> <i>geboortebeperving</i> <sub>CN</sub> (lit: 'method for birth control') 'birth control method'
SN	FC	ADJ	SN	Swedish (  ): <i>brottet</i> <sub>SN</sub> <i>mot</i> <sub>FC</sub> <i>mänsklig</i> <sub>ADJ</sub> <i>rättigheterna</i> <sub>SN</sub> (lit: 'abuses of human rights') 'human rights abuses'
SN	ADJ	FC	SN	Spanish (  ): <i>consumo</i> <sub>SN</sub> <i>final</i> <sub>ADJ</sub> <i>de</i> <sub>FC</sub> <i>energía</i> <sub>SN</sub> (lit: 'consumption final of energy') 'energy end consumption'
ADJ	SN	FC	SN	Danish (  ): <i>gennemsnitlige</i> <sub>ADJ</sub> <i>overførsel</i> <sub>SN</sub> <i>af</i> <sub>FC</sub> <i>data</i> <sub>SN</sub> (lit: 'average transfer of data') 'data transfer rate'

Table 24.4.: Examples of paraphrases for the six selected APPs

- if there is a functional context (FC), the order is: **head**, FC, **modifier** (as in ADJ SN  
FC SN)

The assignment of the structure class relies on the assumption of **cross-lingual head correlation**, i.e., the **head** of the APP correlates with the **head** of the **target compound**. If the **head** of the APP is a complex unit, then the **head** of the **target compound** is also complex (meaning a RIGHT-branched structure) and so for a complex **modifier** or simplex **head** (meaning a LEFT-branched structure).

Exceptions of this assumption are cases of **constituent swapping**, as discussed in Section 5.3.3 and will be addressed in Section 24.3.

## 24.2. Aligned Phrase Pattern Parsing

The **Aligned Phrase Pattern Parsing** (APPP) is a method that determines the **internal structure** of a **ternary target compound** given a set of co-occurring APPs and their corresponding structure classes, as shown in Table 24.3. Based on the structure class assignment, discussed in Section 24.1.3, we define a structure function  $\tau$  which maps an APP to its structure class. If the APP and its structure class are undefined, the class

UNKNOWN is used.

Algorithm 24.1 shows the pseudo-code of the APPP when assuming a cross-lingual head correlation.

---

**Algorithm 24.1** APPP with cross-lingual head correlation assumption

---

**Target:** Expression  $\Psi$

**Input:** APPs of  $\Psi$  for all support languages  $l_i \in L$

```

1: [ ]  $\leftarrow$  Structures
2: for all aligned support languages  $l_i \in L$  do
3:   Structurei  $\leftarrow$   $\tau(\text{APP}_i)$ 
4:   if Structurei  $\neq$  UNKNOWN then
5:     Structures  $\leftarrow$  Structures + [Structurei]
6:   end if
7: end for
8: return max(Structures)

```

---

For all support languages, the structure classes of the corresponding APPs of a target compound  $\Psi$  are collected if they are not UNKNOWN (lines 1-7). Finally, the majority structure class among the collected instances is returned (line 8). In the case of a tie, no prediction is made.

### 24.3. Aligned Phrase Pattern Parsing with Word Alignment Support

The cross-lingual head correlation assumption works for most compounds and their cross-lingual equivalents, but there are cases of constituent swapping, as discussed in Section 5.3.3, i.e., the target compound's head correlates with the APP's modifier and the target compound's modifier correlates with the APP's head. Some examples of constituent swapping are given below.

- |   |  |
|---|--|
| <p>(18) Dutch: <i>stabiële<sub>1</sub> wisselkoersen<sub>2</sub></i><br/>                 stable<sub>1</sub> {exchange rate}<sub>2</sub><br/>                 "{exchange rate}<sub>2</sub> stability<sub>1</sub>"</p> | <p>(19) German: <i>Resolutions<sub>1</sub> entwurf<sub>2</sub></i><br/>                     {resolution<sub>1</sub> draft<sub>2</sub>}<br/>                     "draft<sub>2</sub> resolution<sub>1</sub>"</p> |
|---|--|

For taking into account constituent swapping during APPP, a word alignment-based interpretation of the complex unit is added to the compound parser. Therefore, we define a function  $\text{CU}(\Psi, \zeta)$ , which returns the constituents of the target compound  $\Psi$  that are

---

**Algorithm 24.2** APPP with word alignment interpretation

---

**Target:** Expression  $\Phi$ **Input 1:** APPs of  $\Psi$  for all aligned support languages  $l_i \in L$ 

```

1: []  $\leftarrow$  Structures
2: for all support languages  $l_i \in L$  do
3:   defaulti  $\leftarrow$   $\tau(\text{APP}_i)$ 
4:   if defaulti  $\neq$  UNKNOWN then
5:     if CU( $\Psi$ , APPi)  $\sim$  {B, C} or CU( $\Psi$ , APPi)  $\sim$  {A, C} then
6:       Structures  $\leftarrow$  Structures + [RIGHT]
7:     else if CU( $\Psi$ , APPi)  $\sim$  {A, B} then
8:       Structures  $\leftarrow$  Structures + [LEFT]
9:     else if CU( $\Psi$ , APPi)  $\sim$  {A, B, C} then
10:      Structures  $\leftarrow$  Structures {indicates word alignment error}
11:    else
12:      Structures  $\leftarrow$  Structures + [defaulti] {otherwise, use the default}
13:    end if
14:  end if
15: end for
16: return max(Structures)

```

---

aligned to the complex unit of the APP  $\zeta$ . For APPs having an UNKNOWN structure class (i.e., no complex unit),  $\text{CU}(\Psi, \zeta)$  is undefined. Algorithm 24.2 shows the pseudo-code of the revised procedure of APPP when integrating the word alignment interpretation of the complex unit,  $\text{APPP}_{WA}$ .

For all support languages, the default structure class is determined (lines 2-3). If the structure class is known, we consider the constituents of the complex unit of APP<sub>*i*</sub>: if the target constituent sets {B, C} or<sup>1</sup> {A, C} are aligned to the complex unit in the APP, the indicated structure class is RIGHT (lines 5-6). If the target constituent set {A, B} is aligned to the complex unit in the APP, the indicated structure class is LEFT (lines 7-8). If all three target constituents (i.e., A B C) are aligned to the complex unit in the APP, this is an indicator for a word alignment error. In this case, there is no structure added to the structure list (lines 9-10). In all other cases, the default structure class of the APP (as given in Table 24.3) is used (lines 11-12). As for regular APPP, in the final step, the majority structure class among the collected instances is returned (line 16). In the case of a tie, no prediction is made.

---

<sup>1</sup>In  $\text{APPP}_{WA}$ , we combine two basic approaches of compound parsing: the adjacency model (AdjMod) and the dependency model (DepMod), which have been described in Section 23.1.

## 24.4. Experiment

### 24.4.1. Dataset

As grounding for [parsing ternary compounds](#), we used the CCR(0)-variant of the ENCD, because alignments to [closed compounds](#) (which happen more often for  $\Xi_{closed} > 0$ ) are not of interest for extracting expressive [APPs](#) and with increasing  $\Xi_{closed}$  we would have fewer expressive [APPs](#) among the [closed compounding languages](#).

### 24.4.2. Gold Standard Annotation

Two trained human annotators (including the author of this thesis) individually labeled a sample of 100 randomly selected [3NCs](#). Although, the annotation of [3NCs](#) was [token-based](#), we restricted the sampling to a set of 100 unique [3NCs](#), ensuring a greater variety in our selection. For each [3NC](#) sample, the annotators were given the accompanying sentence. This context helped the annotators to disambiguate the structure of the [3NC](#) (in the case of a context-dependent structural ambiguity).

Since the annotators were no domain experts and [terms](#) in EUROPARL can be quite domain-specific, they were allowed to look up the meaning of the [constituents](#) in a dictionary or check Google for a description.

The annotators were asked to label [3NCs](#) as LEFT, RIGHT, UNKNOWN (if they were unclear) or ERROR (in the case of an extraction error, e.g., due to [PoS](#) errors). This leads to a set of 76 [compounds](#) labeled either as LEFT or RIGHT by both annotators.

The [Inter-Annotator Agreement \(IAA\)](#) rate was 89% with a  $\kappa$  score of 0.693 (Cohen, 1960), which means substantial agreement (Landis and Koch, 1977).

In the next step, both annotators discussed the disagreements and revised their annotations afterwards. This leads to a perfect [IAA](#).

There are 7 samples labeled as UNKNOWN by both annotators. This shows that in some cases, the structure of a [3NC](#) remains ambiguous even in context. One reason for an UNKNOWN label is the phenomenon of [semantic indeterminacy](#), i.e., semantic equivalence of distinct structures.

### 24.4.3. Methods in Comparison

In this pilot study, we do not aim to develop a [compound parsing](#) approach which is competitive with monolingual state-of-the-art methods, but to find evidence for the potential of [cross-lingual](#) support for [compound parsing](#). While we expect a solid accuracy,

we are aware of the fact that relying on APPs is very restrictive and leads to a low non-competitive coverage.

Therefore, we do not compare the presented approaches against state-of-the-art methods, but use only the majority class baseline, i.e., a method that always predicts the majority structure class, the **LEFT class baseline**. Outperforming the baseline illustrates the potential of **cross-lingual** evidence for **compound parsing**.

We will compare the **APPP** against knowledge-lean state-of-the-art methods and advanced **cross-lingual** approaches in the next chapter, Section 25.2.3.

#### 24.4.4. Evaluation Measure

As there are only two possible structure classes for **3NCs**, viz., LEFT or RIGHT, we regard this task as a **binary** classification and score the **accuracy** of class agreement.

#### 24.4.5. Results

Table 24.5 shows the results for **parsing** the 76 samples in our gold standard as either LEFT or RIGHT.

Method	Accuracy
LEFT baseline	71.1%
<b>APPP</b>	89.0%👍
<b>APPP<sub>WA</sub></b>	<b>91.6%👍</b>

Table 24.5.: Parsing results for **APPP** and **APPP<sub>WA</sub>**

The **LEFT class baseline** achieves an accuracy of 71.1%. This is in line with observations discussed in previous literature (e.g., Lauer (1994), Table 23.1), saying that about 66% of all **TCs** are LEFT-branched.

When **parsing** the test samples with **APPP**, we achieve an accuracy of 89.0%, which is significantly<sup>2</sup> (👍) better than the **LEFT class baseline**.

The **compound parsing** using **APPs** and **word alignment** information (**APPP<sub>WA</sub>**) leads to the best results with 91.6% accuracy, outperforming both the **LEFT class baseline** and regular **APPP**.

<sup>2</sup>Approximate randomization test (Yeh, 2000),  $p < 5\%$



### 24.4.6. Discussion

The results show that [cross-lingual](#) information can successfully be used for [parsing compounds](#).

However, there are two crucial limitations of the pattern-based [parsing](#) approach.

**Human Support.** The method depends on a set of predefined [APPs](#), i.e., human support is necessary. While the set of [APPs](#) for [parsing 3NCs](#) can be used as listed in Table 24.3, when switching to a higher arity (e.g., [4NCs](#)), the number of possible [APPs](#) will become larger and the [APPs](#) will be more complex (e.g., a balanced tree structure with four [constituents](#) could be determined using the [APP CN FC CN](#) as in the German *Aktionspläne<sub>CN</sub> für<sub>FC</sub> Energieeffizienz<sub>CN</sub>* ‘energy efficiency action plans’). A possible solution could be the automatic determination of [APPs](#) and their assigned structure classes using a semi-supervised technique (cf. [semantic lexicon bootstrapping](#)). This extension will be addressed in future work (see Section 26.3.1).

**Coverage.** In some cases, a [nominal compound](#) is not translated as a phrase but the [constituent equivalents](#) are spread across the aligned sentence or are separated with external content as in *human<sub>A</sub> rights<sub>B</sub> violations<sub>C</sub>* aligned to the Italian sentence fragment *che le violazioni<sub>C</sub> gravi e sistematiche dei diritti<sub>B</sub> umani<sub>A</sub>*. This leads to a low coverage, because [3NCs](#) being aligned to such constructions cannot be [parsed](#) due to the lack of expressive [APPs](#). However, actually, the example of the Italian translation is expressive, as the [equivalent](#) of the last noun *violazioni* ‘violations’ is linearly separated from the rest *diritti umani* ‘human rights’. In Chapter 25, we will present another [cross-lingual compound parsing](#) approach which does not rely on predefined [APPs](#) but on a [cross-lingual](#) metric that exploits the sentence position of the [constituent equivalents](#) of a [target compound](#).

*24. Pilot Study using Aligned Phrase Patterns*

# 25. Compound Parsing Methods using Aligned Word Distance

In this chapter, we present and elaborate parts of the work published in Ziering and Van der Plas (2015a) and Ziering and Van der Plas (2015b).

While the pattern-based [parsing](#) approaches [APPP](#) and [APPP<sub>WA</sub>](#), presented in Chapter 24, rely on a hand-crafted and fixed set of predefined [APPs](#) and thus cannot make a [structural analysis](#) for [compounds](#) aligned to UNKNOWN or unexpressive [APPs](#), the [parsing](#) methods in this chapter are based on a [cross-lingual](#) metric and can generalize over [APPs](#), which overcomes the necessity of human support in compiling [APPs](#) and leads to a higher coverage of structure predictions.

## 25.1. Aligned Word Distance

The core of all presented [parsing](#) methods in this chapter is a metric for measuring the [semantic association](#) of the [constituents](#) of a [target compound](#). According to our [guiding principle](#) (Section 22.1.2), spatial proximity correlates with [semantic association](#). For getting a well-defined measure of spatial proximity that can be used in [cross-lingual compound parsing](#), we propose the metric of [aligned word distance](#) (AWD). The AWD between two [target constituents](#) is the minimum [word distance](#) between the [constituent equivalents](#).

Given a [support language](#)  $l$  and a [target compound](#)  $\Psi$ , we define the [aligned word set](#) (AWS) of a [target constituent](#)  $c_i \in \Psi$ ,  $\text{AWS}_l(c_i)$ , as the set of sentence position-aware aligned [content words](#)<sup>1</sup> of  $c_i$  (i.e., the [cross-lingual equivalents](#)) in  $l$ . If  $c_i$  is a complex [constituent](#) composed of the [atomic constituents](#)  $c_{i_1}, \dots, c_{i_n}$ ,  $\text{AWS}_l(c_i)$  is the union of AWSs of all [atomic constituents](#) of  $c_i$ , as given in Formula 25.1, where  $\Rightarrow_l$  denotes the [word alignment](#) relationship from the [target language](#) to the [support language](#)  $l$  and

---

<sup>1</sup>We assume that the [target constituents](#) are exclusively aligned to [content words](#), and thus for avoiding noise due to [word alignment](#) errors, we remove any [function words](#) from the [aligned word sets](#).

## 25. Compound Parsing Methods using Aligned Word Distance

$pos(w_i)$  is the sentence position (e.g., a **token** counter starting by 1) of an aligned **word**  $w_i$ .

$$AWS_l(c_i) = \begin{cases} \{w_{i,x} | \text{content word}(w_{i,x}) \wedge pos(w_i) = x \wedge c_i \Rightarrow_l w_i\} & \text{if } c_i \text{ is atomic} \\ \bigcup_{c_{i_k} \in c_i} AWS_l(c_{i_k}) & \text{if } c_i \text{ is complex} \end{cases} \quad (25.1)$$

For two **target constituents**  $c_i, c_j \in \Psi$ , the  $AWD_l$  between them is defined as given in Formula 25.2.

$$AWD_l(c_i, c_j) = \min_{x \in AWS_l(c_i), y \in AWS_l(c_j)} |pos(x) - pos(y)| \quad (25.2)$$

The  $AWD$  considers the minimum pairwise distance between all aligned **words** of both **target constituents**. The  $AWD$  of two **constituents** being aligned to the same aligned **word** (e.g., to a **closed compound**) is zero, indicating the strongest **semantic association** that is measurable using  $AWD$ .

For example, considering the **3NC** *human rights violations* being aligned to the Italian sentence fragment  $\dots che_1 le_2 \mathbf{violazioni}_3 gravi_4 e_5 sistematiche_6 dei_7 \mathbf{diritti}_8 \mathbf{umani}_9$ , the  $AWS_{it}(human)$  is  $\{umani_9\}$ , the  $AWS_{it}(rights)$  is  $\{diritti_8\}$  and the  $AWS_{it}(violations)$  is  $\{violazioni_3\}$ , where the (relative) sentence position is given as subscript. For getting the pairwise  $AWDs$  of the **target constituents**, we can calculate the three values as shown below:

$$\begin{aligned} AWD_{it}(human, rights) &= \min_{x \in AWS_{it}(human), y \in AWS_{it}(rights)} |pos(x) - pos(y)| \\ &= \min_{x \in \{umani_9\}, y \in \{diritti_8\}} |pos(x) - pos(y)| \\ &= |9 - 8| = 1 \end{aligned}$$

$$\begin{aligned} AWD_{it}(human, violations) &= \min_{x \in AWS_{it}(human), y \in AWS_{it}(violations)} |pos(x) - pos(y)| \\ &= \min_{x \in \{umani_9\}, y \in \{violazioni_3\}} |pos(x) - pos(y)| \\ &= |9 - 3| = 6 \end{aligned}$$

$$\begin{aligned}
\text{AWD}_{it}(\text{rights}, \text{violations}) &= \min_{x \in \text{AWS}_{it}(\text{rights}), y \in \text{AWS}_{it}(\text{violations})} |\text{pos}(x) - \text{pos}(y)| \\
&= \min_{x \in \{\text{diritti}_8\}, y \in \{\text{violazioni}_3\}} |\text{pos}(x) - \text{pos}(y)| \\
&= |8 - 3| = 5
\end{aligned}$$

The values for the AWDs indicate that using *Italian* as **support language**, the strongest **semantic association** is between the **target constituents** *human* and *rights*, pointing to a LEFT-branched 3NC.

The AWD is designed in a flexible way that allows for complex **constituents**. For example, in the 4NC *exhaust gas purification technology*, it is possible to calculate the AWD between *exhaust* and *gas* as well as between *exhaust gas* and *purification* or *purification technology*.

As final remark, it needs to be said that our guiding principle cannot be used for predicting an equivalence in **semantic association**. For example, a **closed 3NC**, where all **constituent** pairs have a zero AWD, does not mean that all **constituent** pairs have the same **semantic association**. Therefore, equal AWDs are treated as providing no evidence for the **internal structure** of the **target compound** in the following **cross-lingual compound parsers**.

## 25.2. Deterministic Bottom-Up Parsing

Deterministic Bottom-Up Parsing (DBUP) represents an unsupervised **parser** that starts bottom up with **atomic target constituents** and iteratively merges two adjacent **constituents** until there is only one **constituent** left, which comprises the entire **target compound**.

### 25.2.1. The Algorithm

Algorithm 25.1 shows the pseudo-code for DBUP.

The DBUP algorithm is applied for each **support language**  $l_j$  separately. The input is a list of  $k$  **atomic target constituents** of a **target compound**  $\Psi$ , stored as BOTTOM. The final list of mediate and **immediate constituents** of the resulting **parse** are stored in the list CONs, which serves for constructing the final **parse tree**. This list is initialized with all **atomic target constituents** from BOTTOM (line 1). As long as there is more than one

**Algorithm 25.1** Deterministic Bottom-Up Parsing

---

**BOTTOM:** initialized with the **atomic constituents** of a **target compound**  $\Psi$ :  $c_1, \dots, c_k$   
**Input:**  $AWS_{l_j}(c_i)$  for the **support language**  $l_j$  and for all **constituents**  $c_i \in \text{BOTTOM}$

- 1:  $\text{CONs} \leftarrow c_1, \dots, c_k$       {**constituent** list for constructing the final **parse tree**}
- 2: **while**  $|\text{BOTTOM}| > 1$  **do**
- 3:     $(c_m, c_{m+1}) \leftarrow$  determine the pair of adjacent **constituents** with the smallest **AWD**
- 4:     $AWS_{l_j}(c_{[m,m+1]}) = AWS_{l_j}(c_m) \cup AWS_{l_j}(c_{m+1})$       {the two **AWSs** are unified}
- 5:    replace  $c_m$  and  $c_{m+1}$  in **BOTTOM** by  $c_{[m,m+1]}$       {the two **constituents** are merged}
- 6:     $\text{CONs} \leftarrow \text{CONs} + c_{[m,m+1]}$       {the merged **constituent** is added to **CONs**}
- 7: **end while**
- 8: **return** **parse tree**(**CONs**)      {the final **parse tree** is constructed using **CONs**}

---

**constituent** in **BOTTOM** (line 2), in each iteration, the pair of adjacent **constituents** having the smallest **AWD** is determined (line 3). For this pair, the corresponding **AWSs** are unified (line 4) and the **constituents** are replaced by the merged (complex) **constituent** in **BOTTOM** (line 5). Finally, the merged **constituent** is stored in **CONs** (line 6). The output of the algorithm is a **parse tree** which is constructed from **CONs** (line 8).

If the smallest **AWD** is not unique but the **target constituents** under consideration do not overlap (e.g.,  $(c_1, c_2)$  and  $(c_3, c_4)$  are aligned to two different support **closed compounds**), both **target constituent** pairs are merged in one iteration. If the **target constituents** overlap (e.g.,  $(c_1, c_2)$  and  $(c_2, c_3)$  are aligned to a common support **closed compound**), no **parse tree** can be derived from the **support language**. Similarly, if there is an empty  $AWS(c_i)$ , i.e., there is no alignment from  $c_i$  to a **content word**, **DBUP** cannot produce a **parse tree** using the **support language**.

After having applied **DBUP** to all **support languages**, as final **parse tree** the majority vote of all collected **parse trees** is used. In the case of a tie, **DBUP** does not produce a final structure.

An alternative strategy for cases of tie (which is not taken in the experiments presented in Section 25.2.3) is to provide all top-ranked **parse trees** or a frequency rank of all **parse trees**. Such a non-deterministic output can be considered two-fold: (1) providing several **parse trees** with the same top-rank is an indicator for **semantic indeterminacy** and (2) a back-off model on **compound parsing** can be applied to a narrowed search space (e.g., two instead of five possible **parse trees** for a **4NC**).

**DBUP** can be applied **token-based** or **type-based**. For the **token-based** mode, only the aligned sentences of a certain instance of the **target compound**  $\Psi$  are considered, whereas for the **type-based** version, all translations from all instances of  $\Psi$  available in the **ENCD** are taken into account.

### 25.2.2. Example cases

#### *air transport safety organization*

A first example for illustrating the procedure of DBUP is the 4NC *air transport safety organization* aligned to four words in the French sentence fragment *Nous devons mettre en place cette **organisation**<sub>7</sub> européenne chargée de la **sécurité**<sub>12</sub> du **transport**<sub>14</sub> **aérien**<sub>15</sub> qui ...* ‘We need to establish this European **organization** responsible for the **safety** of **air transport** that ...’.

In this scenario,  $AWS_{fr}(air)$  is  $\{aérien_{15}\}$ ,  $AWS_{fr}(transport)$  is  $\{transport_{14}\}$ ,  $AWS_{fr}(safety)$  is  $\{sécurité_{12}\}$  and  $AWS_{fr}(organization)$  is  $\{organisation_7\}$ .

The target constituents  $c_1$  (*air*) and  $c_2$  (*transport*) have the smallest AWD and thus are merged first. In the next iteration, the smallest AWD is between  $c_{[1,2]}$  (*air transport*) and  $c_3$  (*safety*). As last step, we merge  $c_{[[1,2],3]}$  (*air transport safety*) and  $c_4$  (*organization*).

The resulting left-peripheral parse tree is shown in Figure 25.1.

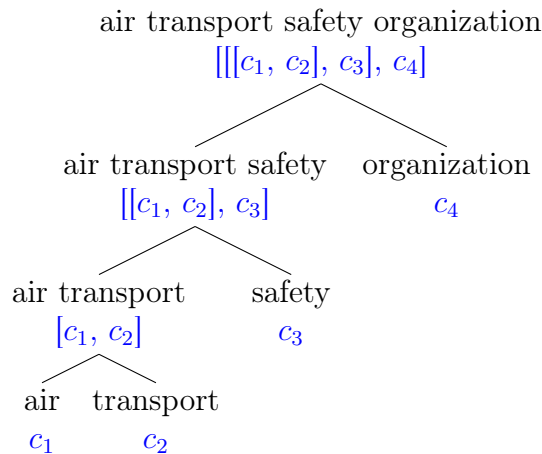


Figure 25.1.: DBUP parse tree for *air transport safety organization*

#### *twin pipe undersea gas pipeline*

Another illustrative example is the 5NC *twin pipe undersea gas pipeline* being aligned to four words in the Dutch sentence fragment *gaat de langste **onderzeese**<sub>21</sub> **gaspijpleiding**<sub>22</sub> met **dubbele**<sub>24</sub> **pijp**<sub>25</sub> ter wereld worden* ‘is the longest undersea gas pipeline with double pipe in the world’.

In this scenario,  $AWS_{nl}(twin)$  is  $\{dubbele_{24}\}$ ,  $AWS_{nl}(pipe)$  is  $\{pijp_{25}\}$ ,  $AWS_{nl}(undersea)$  is  $\{onderzeese_{21}\}$ ,  $AWS_{nl}(gas)$  is  $\{gaspijpleiding_{22}\}$  and  $AWS_{nl}(pipeline)$  is also  $\{gaspijpleiding_{22}\}$ .

The smallest [AWD](#) is between  $c_4$  (*gas*) and  $c_5$  (*pipeline*). In the next iteration, the smallest [AWD](#) is both between  $c_1$  (*twin*) and  $c_2$  (*pipe*), and between  $c_3$  (*undersea*) and  $c_{[4,5]}$  (*gas pipeline*). Both [target constituent](#) pairs are merged in one step. As last iteration,  $c_{[1,2]}$  (*twin pipe*) and  $c_{[3,[4,5]]}$  (*undersea gas pipeline*) are merged.

The resulting [parse tree](#) is shown in [Figure 25.2](#).

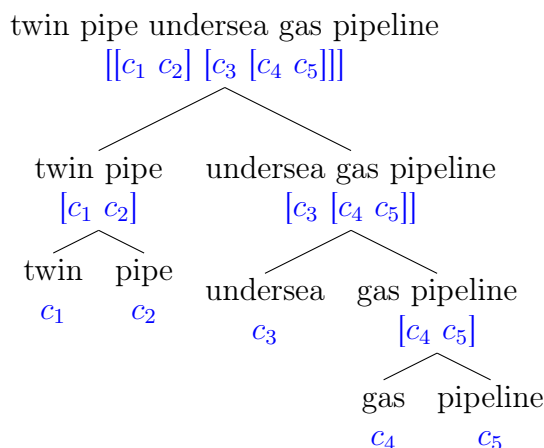


Figure 25.2.: DBUP parse tree for *twin pipe undersea gas pipeline*

### 25.2.3. Experiments

#### Data

Similarly to our pilot study in [Chapter 24](#), we used the [CCR\(0\)](#) version of the [ENCDC](#) as grounding source for sampling our test [compounds](#).

As exemplified in [Section 25.2.2](#), [DBUP](#) is applicable to [compounds](#) with any [compound size](#) (in terms of [atomic constituents](#)). However, in several cases, the [cross-lingual](#) context does not suffice to get a full [parse](#) of a [kC](#) ( $k > 3$ ). The combination of partial results will be addressed in [Section 25.3](#). Moreover, we want to compare [DBUP](#) with previous work on [compound parsing](#) which focus on classifying [3NCs](#) as either LEFT- or RIGHT-branched. Thus, we decided to restrict to the largest class of complex [compounds](#), viz., [3NCs](#) (93.8% of all [compounds](#) with three or more [constituents](#) in the [ENCDC](#), [CCR\(0\)](#)). In contrast to previous work (e.g., [Vadas and Curran \(2007b\)](#)) we take only common nouns as [constituents](#) into account rather than named entities. We consider the task of [parsing 3NCs](#) composed of common nouns more ambitious, because named entities often form a single concept that is easy to spot, e.g., *Apple II owners*.



Thus, from our grounding source, we extract only entries whose **PoS pattern** matches with a sequence of three common nouns.

Extraction errors are a problem, since many adjectives have been tagged as nouns and some **3NCs** occur as incomplete fragments. For increasing the effectiveness of human annotation, we developed a high-confidence noun filter  $P_{noun}(word) = P(noun | word)$ . It is trained on the English WIKIPEDIA tagged by TREETAGGER (Schmid, 1995). We inspect all **3NCs** in the context of one **token** to the left and right,  $w_0\{N_1N_2N_3\}w_4$ . If  $P_{noun}(N_i) < \theta$  or  $P_{noun}(w_j) \geq \theta$ , we remove the **3NC** from our dataset. By inspecting a subset of all **3NCs** in the **ENCD**, we estimated the best filter quality to be with  $\theta = 0.04$ . For example, this threshold discards *increasing land abandonment* but keeps *human rights abuse*. Our final dataset contains 14,941 **3NC tokens** and 8824 **3NC types**.

### Gold Standard Annotation

For comparing **DBUP** against previous work, the test set of 76 **compounds** used in our pilot study (Chapter 24) is too small. Moreover, we want to have more classes for labeling **3NCs**. Therefore, we decided to create a new test set for **DBUP**.

A trained independent annotator classified a randomly selected set of 1100 **compound tokens**<sup>2</sup> accompanied with the surrounding sentence with one of the following labels:

**LEFT:** the **3NC** is LEFT-branched

**RIGHT:** the **3NC** is RIGHT-branched

**ERROR:** for falsely extracted **compounds** that survived the high-confidence noun filter  $P_{noun}(word)$

**UNKNOWN:** the **3NC** cannot be disambiguated within the one-sentence context

**SEMIND:** the **3NC** is **semantically indeterminate**, i.e., LEFT and RIGHT have the same meaning as in *book price fixing* (i.e., *price fixing for books* vs. *fixing of the book price*)

Two additional trained independent annotators each classified one half of the dataset for checking **Inter-Annotator Agreement (IAA)**. For the classes LEFT and RIGHT (308 **compound tokens**), we achieved an **IAA** rate of 90.3% and  $\kappa = 0.717$  (Cohen, 1960), which means good agreement (Landis and Koch, 1977).

<sup>2</sup>For keeping the annotation effort small, we only sample **3NCs** that can be processed by **DBUP**, i.e., for which there are enough **cross-lingual** cues.

As final test set, we used the LEFT/RIGHT consensus, comprising 278 **compound tokens**.

### Evaluation Measures

We measure the LEFT/RIGHT classification accuracy ( $Acc_{\Omega}$ ) for a set of **3NC tokens**  $\Omega$  as shown in Formula 25.3, i.e., the number of correctly LEFT- or RIGHT-branched **3NCs** divided by size of  $\Omega$ .

$$Acc_{\Omega} = \frac{freq(\text{LEFT}\checkmark) + freq(\text{RIGHT}\checkmark)}{|\Omega|} \quad (25.3)$$

The coverage ( $Cov$ ) of a method is shown in Formula 25.4, i.e., the number of all assigned LEFT/RIGHT labels divided by all **3NC tokens** in the full dataset described above.

$$Cov = \frac{freq(\text{LEFT}) + freq(\text{RIGHT})}{14,941} \quad (25.4)$$

We consider an optimal **compound parsing** method to have the best trade-off between  $Acc_{\Omega}$  and  $Cov$ . Therefore, we used the harmonic mean (**harmonic**) between these measures.

### Methods in Comparison

We compare **DBUP** with the pattern-based **parsing** approach **APPP**, presented in the pilot study (Chapter 24).

For both **DBUP** and **APPP**, we use the majority structure vote across all nine aligned **support languages**. We show results for both a **token-** and **type-**based interpretation of the **target compounds**.

We implemented an unsupervised statistical **compound parsing** method based on bi-gram frequencies derived from the English part of EUROPARL. As statistical metric for measuring the **semantic association** between the **target constituents**, we used the **Chi Squared** ( $\chi^2$ ) measure, which worked best in previous work on **compound parsing** (Nakov and Hearst, 2005). In this statistical approach, for a **target compound** **A B C**, we compare the **semantic association** between **A** and **B** with that between **B** and **C** (i.e., we use the **AdjMod**), because we observed that it provides better performance than comparing the **semantic association** between **A** and **B** with that between **A** and **C** (i.e., the **DepMod**), as suggested by Lauer (1994).

We defined two back-off models for **APPP** and **DBUP** that back off to using the statistical  $\chi^2$  method if no **parse tree** can be constructed using **cross-lingual** support. We refer to this back-off model as **APPP** $\rightarrow\chi^2$  and **DBUP** $\rightarrow\chi^2$ , respectively.

Finally, we compare with the **LEFT class baseline**.

## Results

Table 25.1 presents the coverage of each system.

System	Coverage
<b>DBUP</b> <sub>token</sub> / <b>DBUP</b> <sub>type</sub>	87.9% / 91.2%
<b>DBUP</b> <sub>type</sub> $\rightarrow\chi^2$	100%
$\chi^2$	100%
<b>APPP</b> <sub>token</sub> / <b>APPP</b> <sub>type</sub>	29.9% / 48.1%
<b>APPP</b> <sub>type</sub> $\rightarrow\chi^2$	100%
<b>LEFT class baseline</b>	100%

Table 25.1.: Parsing coverage for **DBUP** and systems in comparison

Our first result is that **type-based cross-lingual parsing** methods outperform the **token-based** counterparts and achieve up to 91.2% in coverage (**DBUP**<sub>type</sub>). As expected, our pattern-based approach does not cover more than 48.1% (**APPP**<sub>type</sub>). The statistical  $\chi^2$  method and the back-off models can process all **3NCs** in the dataset. The fact that **DBUP**<sub>type</sub> misses 8.8% of the dataset is mainly due to equal **AWDs** between the **target constituents**. For example, *crisis<sub>A</sub> resolution<sub>B</sub> mechanism<sub>C</sub>* is only aligned to **closed compounds**, such as the Swedish *krislösningsmekanism* (i.e.,  $AWD_{sv}(A,B) = AWD_{sv}(B,C) = 0$ ), or to nouns separated by one preposition, such as the Spanish *mecanismo de resolución de crisis* ‘mechanism of resolution of crisis’ (i.e.,  $AWD_{es}(A,B) = AWD_{es}(B,C) = 2$ ).

Since many systems in Table 25.1 do not have full coverage, for a fair comparison (where we do not count an uncovered **compound** as falsely **parsed**) we need to define test subsets that are processable for a group of systems in comparison. Table 25.2 directly compares the systems on common test subsets (*com*), i.e., on sets of **3NCs** for which all systems in the group provide a result (i.e., **LEFT** or **RIGHT**).

The main reason why a **cross-lingual compound parser** predicts the false structure class is the quality of automatic **word alignment**. **DBUP** outperforms **APPP** significantly<sup>3</sup>

<sup>3</sup>Approximate randomization test (Yeh, 2000),  $p < 5\%$

## 25. Compound Parsing Methods using Aligned Word Distance

System	Acc <sub>com</sub>	harmonic( <i>com</i> )	<i>com</i>
DBUP <sub>token</sub> / DBUP <sub>type</sub>	94.4% / 94.4%	91.0% / <b>92.8%</b>	270
APPP <sub>token</sub> / APPP <sub>type</sub>	<b>87.8%</b> / 87.2%	44.6% / <b>62.0%</b>	180
DBUP <sub>type</sub>	<b>94.6%</b> 👍	<b>92.9%</b> 👍	184
APPP <sub>type</sub>	86.4%	61.8%	
DBUP <sub>type</sub>	<b>94.1%</b> 👍	92.6%	273
$\chi^2$	87.9%	<b>93.6%</b>	
DBUP <sub>type</sub> $\rightarrow \chi^2$	<b>93.5%</b> 👍	<b>96.6%</b> 👍	278
APPP <sub>type</sub> $\rightarrow \chi^2$	86.7%	92.9%	
$\chi^2$	87.4%	93.3%	
LEFT class baseline	80.9%	89.4%	

Table 25.2.: Parsing results for **DBUP** and systems in comparison on common test subsets

(👍). This can be explained with the flexible structure of **DBUP**, which can exploit more data and is thus more robust to **word alignment** errors. **DBUP** significantly (👍) outperforms  $\chi^2$  in accuracy but is inferior in **harmonic(*com*)**; here, the higher coverage of  $\chi^2$  outweighs its poorer accuracy. The last group in Table 25.2 shows all systems with a full coverage. **DBUP**'s back-off model achieves the best **harmonic(*com*)** with **96.6%** and an accuracy comparable to human performance.

For **DBUP**, **types** and **tokens** show the same accuracy (94.4%). In contrast, for **APPP** the **token**-based approach is superior to the **type**-based variant. However, the **harmonic(*com*)** numbers for **APPP** illustrate that coverage gain of **types** outweighs the higher accuracy of **tokens**. Our general intuition that **token**-based approaches are superior in accuracy is hardly reflected in the present results. We believe that this is due to the domain-specificity of EUROPARL: there are only very few instances, where the structure of a **3NC** differs from **token** to **token**. We expect to see a larger accuracy difference for general domain **parallel corpora**. The application of our **cross-lingual compound parsing** methods to such corpora will be addressed in future work (see Section 26.3.6).

Language family	Acc <sub>com</sub>	Cov	harmonic( <i>com</i> )	<i>com</i>
Romance	86.6%	<b>86.2%</b>	<b>86.4%</b>	201
Germanic	<b>94.0%</b>	68.0%	78.9%	

Table 25.3.: Comparison of different language families for **type**-based **DBUP**

Table 25.3 shows the contribution of the Romance (i.e., *French, Italian, Portuguese* and *Spanish*) and Germanic **support languages** (i.e., *Danish, Dutch, German* and *Swedish*) for  $\text{DBUP}_{type}$ . The first observation is that Romance **support languages** have a higher coverage than Germanic ones (86.2% vs. 68.0%). This is because many **3NCs** are aligned to a **closed compound** in the Germanic **closed compounding languages**, which provides no information on the **internal structure**. Since **cross-lingual equivalents** in Romance **support languages** are usually multiword **complex nominals**, coverage is higher. Our second observation is that Romance **support languages** are worse in accuracy than Germanic languages (86.6% vs. 94.0%). One reason for this is that there is a construction in Romance that violates our guiding principle (22.1.2), viz., the **APP**  $\text{SN} \quad \text{FC} \quad \text{SN} \quad \text{ADJ}$  as in the English **3NC** *state health service* being aligned to the Portuguese *serviços de saúde estatais* (lit.: [*service<sub>SN</sub> of<sub>FC</sub> health<sub>SN</sub> state<sub>ADJ</sub>*]). As we discussed in Section 24.1.2, we neglect this **APP** in our pilot study due to structural ambiguity. However, for being most manual-resource-lean, we avoided to exclude **APPs** in the metric-based **compound parser**. Moreover, we observed that excluding test samples having a Romance **equivalent** matching this **APP** would even worsen the overall performance of **DBUP**. In a follow-up experiment, we observed that test set samples with this **APP** have significantly<sup>4</sup> more **LEFT** labels than the total test set. Furthermore, many instances of these cases can be disambiguated using morphosyntactic information such as number, e.g., the English **3NC** *world fishing quotas* aligned to the French phrase *quotas<sub>s</sub> de pêche mondiaux<sub>s</sub>* (lit: ‘quotas<sub>pl</sub> of fishing<sub>sg</sub> world<sub>adj,pl</sub>’).

#### 25.2.4. Discussion and Conclusion

In this section, we presented the **DBUP** method, a metric-based **cross-lingual compound parser** that iteratively merges adjacent **constituents** with the smallest **AWD**, starting bottom-up with **atomic constituents**. This way, **DBUP** is not relying on predefined **APPs**. This flexibility leads to a significantly higher coverage and accuracy, as has been shown in the experiments above (25.2.3).

However, the coverage of **DBUP** is still lower than for statistical approaches, such as the  $\chi^2$  method, that has full coverage. One reason for this is the fact that the aligned **words** for a **target compound** have to be positioned in a way that there is always a unique smallest **AWD** among all pairs of adjacent **target constituents**. This is not the case for a sequence of three or more **constituents** being aligned to a common **closed**

---

<sup>4</sup>z-test for proportions;  $p < 5\%$

**compound**. For example, the English 4NC *book price fixing schemes* is aligned to the German **closed compound** *Buchpreisbindungsregelungen*, which does not help for **parsing** the 4NC, and to the Danish phrase *fastprisordningerne for bøger* ‘price fixing schemes for books’. The Danish translation cannot be used in DBUP for getting a **parse tree**, because the **constituents** *price*, *fixing* and *schemes* are aligned to the common **closed compound** *fastprisordningerne*. However, the fact that the AWD between *book* and *price* is 2, whereas between the other two **constituent** pairs is 0, can provide a partial **parse tree**, as given in Figure 25.3.

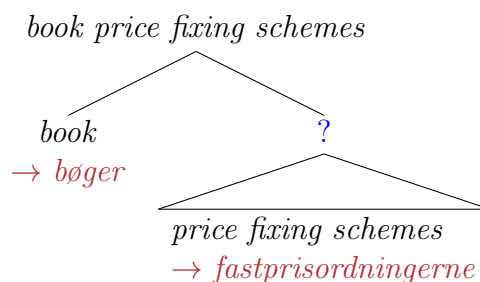


Figure 25.3.: Partial result for parsing *book price fixing schemes* using a Danish phrase

In the next section, we will present a method that is capable of combining partial results of various **support languages** to a final unique **parse tree**.

While applying DBUP to a single **support language** can be considered as a deterministic **parsing** approach (i.e., the method has to decide for a unique **target constituent** pair having the smallest AWD before proceeding), the final step of counting the single **parse trees** of each individual **support language** and determining the predominantly predicted structure can be also done **non-deterministically**: if there is not a unique most frequent **parse tree** or the frequency distribution points to several plausible **parse trees**, DBUP can also produce a list of the most plausible **parse trees** according to the available **cross-lingual** evidence.

## 25.3. Non-deterministic Tree Accumulation Parsing

In this section, we present two methods for **Non-deterministic Tree Accumulation Parsing** (NTAP). NTAP is based on a principle of semantically valid **parse trees**, which will be discussed in Section 25.3.1. In Section 25.3.2, the NTAP focus on full **parse trees**. We call this system consequently the **non-deterministic full tree accumulation parsing** (NFTAP).

In Section 25.3.3, the **NTAP** method is more fine-grained and considers all **subtrees**<sup>5</sup> of a given **parse tree**. That system will be consequently called **Non-deterministic Subtree Accumulation Parsing** (**NSTAP**). Both **NTAP** approaches have their individual advantages as well as benefits compared to **DBUP**, which will be discussed and illustrated with examples.

### 25.3.1. Principle of a Semantically Valid Parse Tree

In analogy to **DBUP**'s process of iteratively merging adjacent **constituents** with the smallest **AWD** from bottom-up, we define a principle of semantic validity of **parse trees** with respect to the **AWD** of the branching **target constituents**, reflecting our **guiding principle** in Section 22.1.2.

A **parse tree**  $PT$  is semantically valid with respect to a **support language**  $l_i$  if for each node  $N \in PT$ , the  $AWD_{l_i}$  between the **target constituents** related to the joined daughter nodes of  $N$  is smaller than (or equal to) the  $AWD_{l_i}$  between the **constituents** related to  $N$  and the sister node of  $N$ .



Figure 25.4.: Possible **parse trees** for *air traffic control*

For example, the **RIGHT**-branched **parse tree** of the **3NC** *air traffic control* (right **parse tree** in Figure 25.4) would not be semantically valid with respect to the Dutch paraphrase *controle van het luchtvaartverkeer* ‘control of air traffic’, because the **target constituents** *air* and *traffic control* have a smaller  $AWD_{nl}$  than *traffic* and *control* have.

The main differences between this principle and the **constituent** merging strategy in **DBUP** is that this principle allows for equal **AWDs** on several levels of a **parse tree**, whereas in **DBUP** the **constituents** to be merged need to have the uniquely smallest

<sup>5</sup>A **subtree**  $st$  of a full tree  $ft$  is a **parse tree** consisting of a node  $N$  of  $ft$  as **root node** and all descendants of  $N$ . This means, the full tree  $ft$  is the largest **subtree**  $st$  of  $ft$ .

**AWD**. Moreover, in **DBUP**, we have a local perspective on the **constituents** at a certain **parse tree** level. As a consequence, differences in the **word** order across languages can make **DBUP** predict a **parse tree** which is not semantically valid according to this principle.

### 25.3.2. Non-deterministic Full Tree Accumulation Parsing

#### The Algorithm

Algorithm 25.2 shows the pseudo-code for **NFTAP**. For a given **target compound**  $\Psi$ , **NFTAP** is applied to each **support language**  $l_i$ , separately.

---

#### Algorithm 25.2 Non-deterministic Full Tree Accumulation Parsing

---

**Target:** Compound  $\Psi$

```

1: Trees  $\leftarrow$  generate all possible binary parse trees for  $\Psi$ 
2: for tree  $t \in$  Trees do
3:   annotate all nodes  $N_i$  in  $t$  with AWDs for the support language  $l_i$ 
4:   if  $\exists N[N.AWD_{l_i} > mother(N).AWD_{l_i}]$  then
5:      $t \leftarrow$  INVALID
6:   end if
7: end for
8: return  $\{t \in$  Trees  $|$   $t$  is not INVALID $\}$ 

```

---

In this method, all possible **binary parse trees** of a **target compound**  $\Psi$  are generated (line 1). As discussed in Section 3.6.4, the number of **binary parse trees** increases with the Catalan numbers (Church and Patil, 1982): **compounds** with  $k$  **constituents** can be represented by  $Cat_{k-1}$  possible **binary** trees, where the formula for  $Cat_n$  is given in Formula 3.1, repeated as Formula 25.5.

$$Cat_n = \frac{(2n)!}{(n+1)! \cdot n!} \quad (25.5)$$

Table 3.2 showed the number of possible **binary parse trees** for **compounds** having up to 15 **constituents**, repeated for up to five **constituents** in Table 25.4.

In the next step, all nodes  $N_i$  in the collected **parse trees** are annotated with  $AWD_{l_i}$  (line 3) according to the formula shown in Formula 25.6, i.e., **leaf nodes** are annotated with  $AWD_{l_i} = 0$  and non-terminal nodes are annotated with the  $AWD_{l_i}$  between their left and right daughters' **constituent**.



Compound size $k$	Binary trees
2	1
3	2
4	5
5	14
$k$	$Cat_{k-1}$

Table 25.4.: Number of possible binary parse trees for compounds with  $k$  constituents

$$N_i.AWD = \begin{cases} \text{leaf node}(N_i) & \mapsto 0 \\ \text{else} & \mapsto AWD(N_i.left, N_i.right) \end{cases} \quad (25.6)$$

All annotated parse trees are validated according to the principle of semantically validity (25.3.1) (lines 4-6), i.e., a parse tree is **valid** if its AWD annotation is monotonically increasing when traversing the tree bottom-up. If there is a node  $N$  whose AWD annotation is greater than the AWD annotation of its mother node, the full parse tree is classified as invalid. Finally, the set of valid parse trees is returned (line 8).

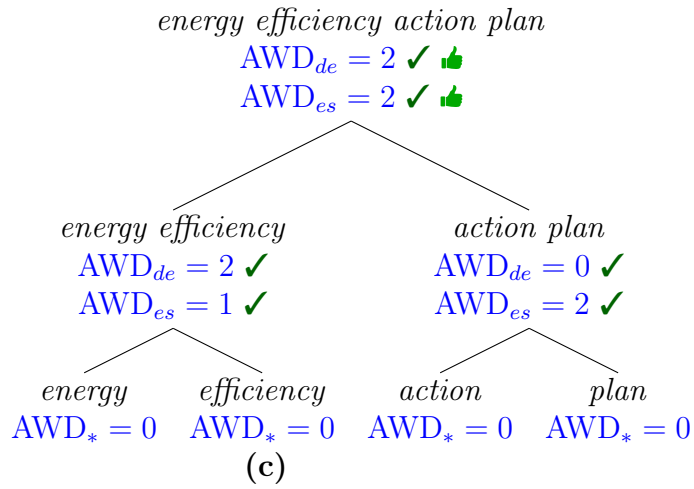
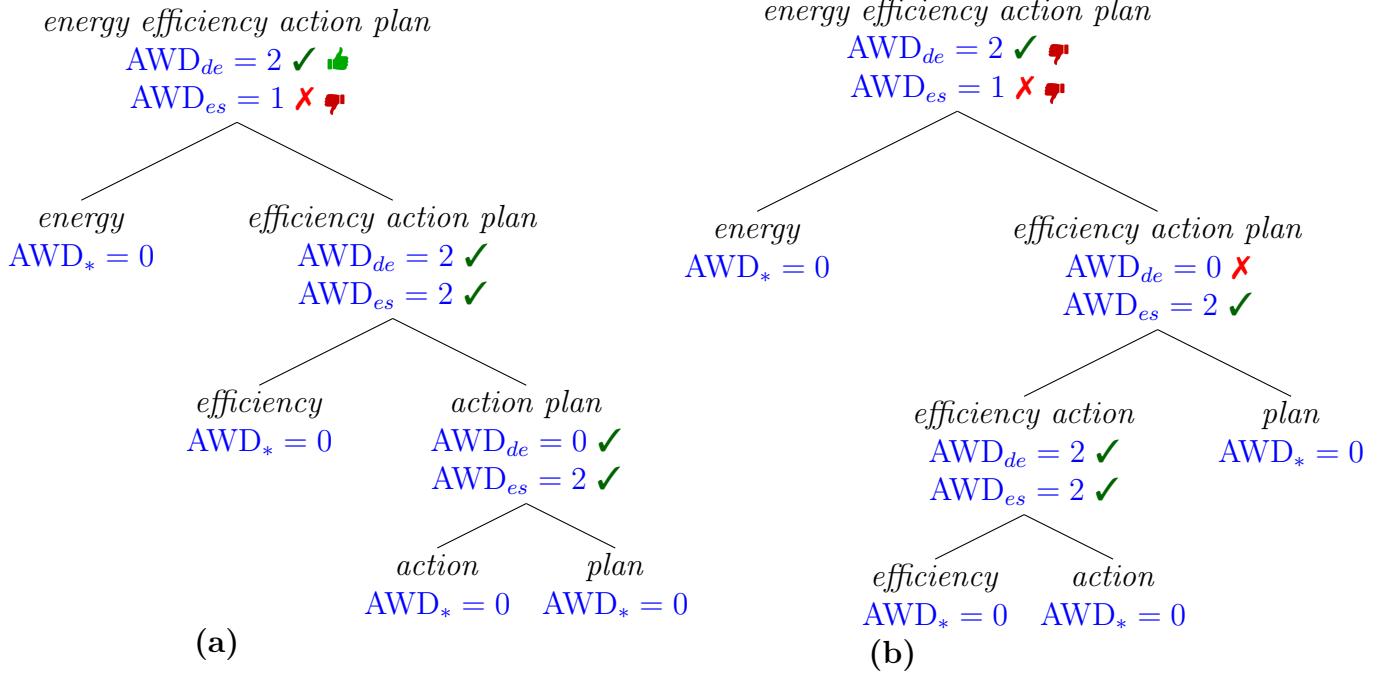
In analogy to DBUP, after having applied NFTAP to all support languages, all valid parse trees returned from each support language are stored in a Full parse Tree Accumulation (FTA). The final parse tree is the majority of all parse trees in the FTA. In the case of a tie, NFTAP provides a set of final parse trees. As discussed for DBUP in Section 25.2.1, this non-deterministic output of NFTAP can have two functions. Firstly, providing several system trees allows for the identification of semantic indeterminacy and secondly, the  $n$ -best list (where implausible parse trees are discarded) can be used by downstream tasks.

In the same way as for DBUP, NFTAP is also applicable token-based (i.e., accumulating the parse trees compatible with aligned sentences of a certain instance of the target compound  $\Psi$ ) and type-based (i.e., accumulating the parse trees which are valid for the aligned sentences of all instances of the target compound  $\Psi$ ).

### Example case - energy efficiency action plan

To illustrate the procedure of NFTAP, we use the 4NC energy efficiency action plan for which there are five possible binary parse trees as shown in Figure 25.5.

For the German equivalent *Aktionsplan zur Effizienz von Energie* ‘action plan



{for the} efficiency of energy’ and the Spanish equivalent *plan de acción de eficiencia energética* ‘plan of action of efficiency energy<sub>adj</sub>’, all five parse trees are annotated with AWDs. Monotonically increasing AWD annotations from a node to its mother node are marked with ✓ (invalid annotations with ✗) at the mother node. A valid full parse tree (i.e., where all AWD annotations are marked with ✓) is marked with 👍 (invalid parse trees with 🚫) at the root node.

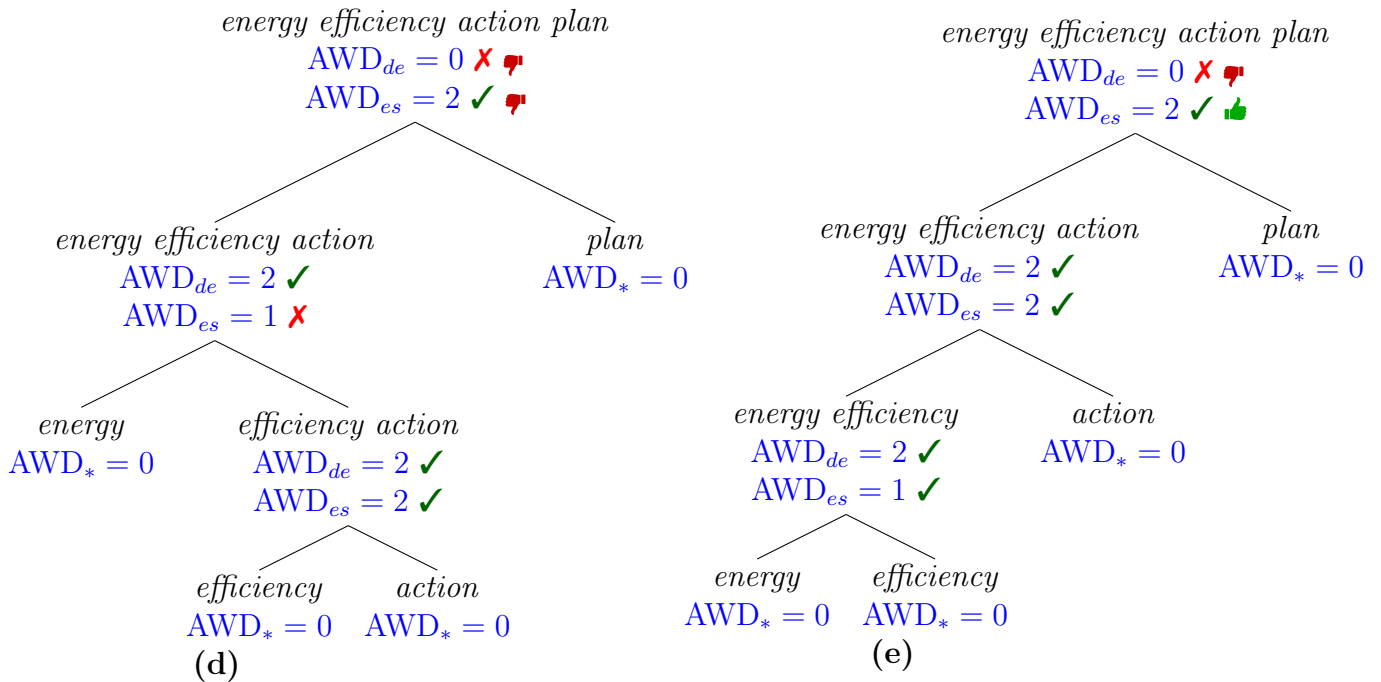


Figure 25.5.: Possible binary parse trees for *energy efficiency action plan*

For the German phrase, the minimum AWD is between *action* and *plan* ( $AWD=0$ ). The AWDs between *energy* and *efficiency*, and between *efficiency* and *action* are both the same ( $AWD=2$ ). Thus, for German there are two valid parse trees in which *action* and *plan* are direct siblings, i.e., the parse trees (a) and (c).

For the Spanish phrase, the minimum AWD is between *energy* and *efficiency* ( $AWD=1$ ). The AWDs between *efficiency* and *action*, and between *action* and *plan* are both the same ( $AWD=2$ ). Thus, for Spanish there are two valid parse trees in which *energy* and *efficiency* are direct siblings, i.e., the parse trees (c) and (e).

When storing all four valid parse tree tokens derived from the German and Spanish equivalents in the FTA, the majority parse tree, the balanced structure (c), where both *action+plan* and *energy+efficiency* are direct siblings, is returned.

**Example case - farm income stabilisation instrument**

For exemplifying the potential of determining semantic indeterminacy with NTAP, we use the 4NC  $farm_A income_B stabilisation_C instrument_D$ , which is semantically indeterminate with respect to two parse trees given in Figure 25.6. This 4NC is aligned to the Danish equivalent  $instrument_D til stabilisering_C af bedrifternes_A indkomster_B$ , to the German equivalent  $Instrument_D zur Stabilisierung_C der landwirtschaftlichen_A Einkommen_B$ , to the Swedish equivalent  $instrument_D för stabilisering_C av jordbruksinkomsterna_{AB}$ , to the Spanish equivalent  $instrumento_D para la estabilización_C de las rentas_B agrícolas_A$ , to the French equivalent  $instrument_D de stabilisation_C du revenu_B agricole_A$ , to the Italian equivalent  $strumento_D di stabilizzazione_C dei redditi_B agricoli_A$ , to the Portuguese equivalent  $instrumento_D de estabilização_C do rendimento_B agrícola_A$  and to the Greek compound equivalent  $εργαλείο σταθεροποίησης γεωργικού εισοδήματος$  (lit: ‘instrument stabilization agricultural income’).

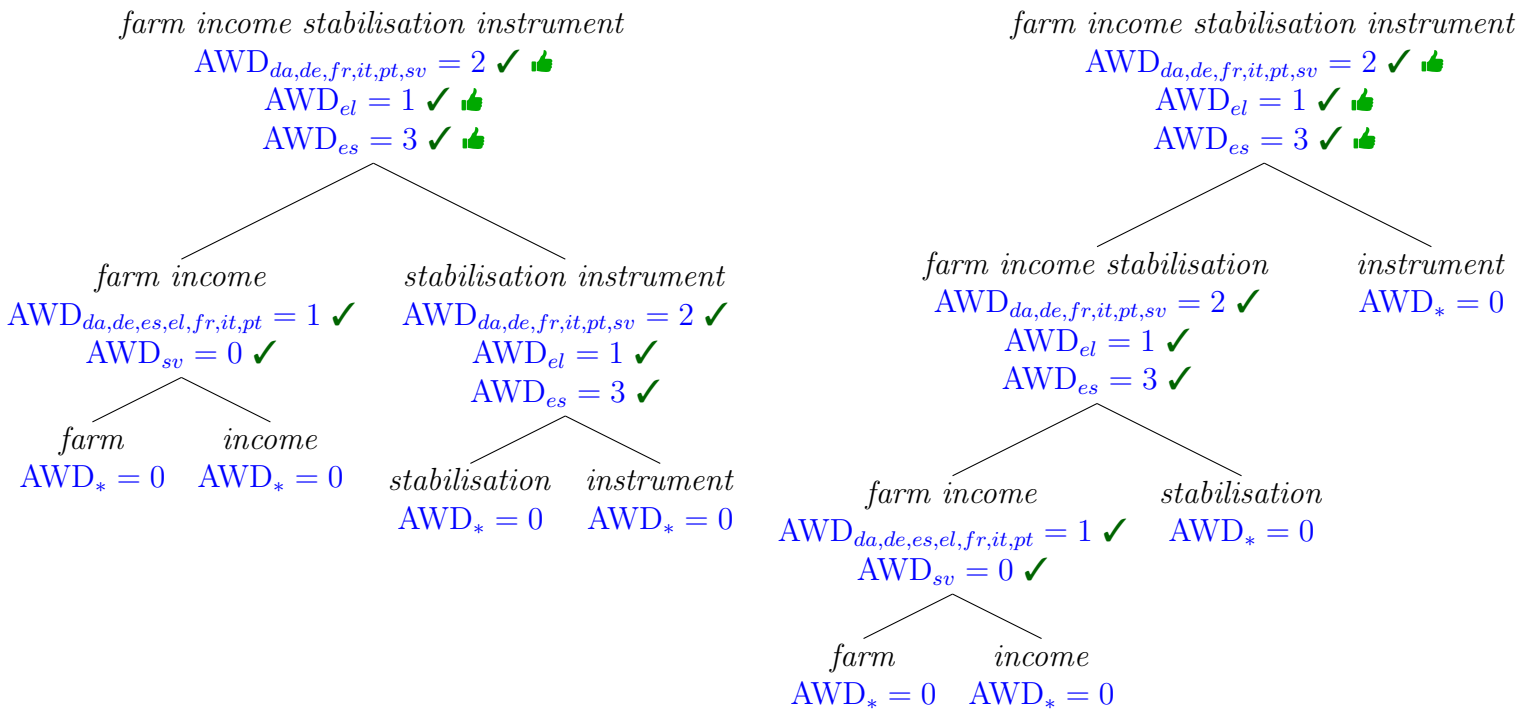


Figure 25.6.: Semantically equivalent trees for *farm income stabilisation instrument*

For all aligned support languages, the AWD between *farm* and *income* is smallest and the AWD between *income* and *stabilisation* equals the AWD between *stabilisation* and *instrument*, leading to the FTA where the semantically equivalent parse trees are top-ranked, as shown in Table 25.5.

Rank	Bracketing	FTA
1	[ farm income ] [ stabilisation instrument ]	8
1	[ [ farm income ] stabilisation ] instrument	8
⋮	...	...

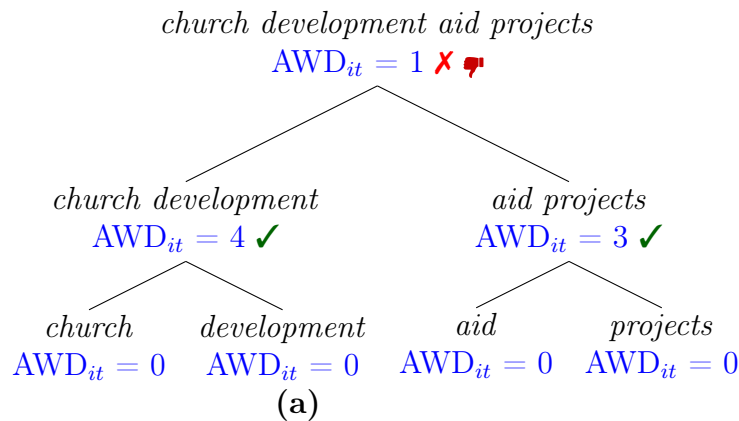
Table 25.5.: FTA for the [semantically indeterminate](#) *farm income stabilisation instrument*

### 25.3.3. Non-deterministic Subtree Accumulation Parsing

#### Problem Description

In many cases, when parsing a [compound](#) using NFTAP, an invalid [parse tree](#) still contains a valuable information in terms of a valid [subtree](#).

As example, we try to [parse](#) the 4NC *church<sub>A</sub> development<sub>B</sub> aid<sub>C</sub> projects<sub>D</sub>*, which has five possible [parse trees](#), shown in Figure 25.7.



When using the Italian phrase *progetti<sub>D</sub> ecclesiastici<sub>A</sub> di aiuti<sub>C</sub> allo sviluppo<sub>B</sub>* (lit: ‘projects ecclesiastical of aid to development’), it is not possible to derive any valid [parse tree](#), as illustrated with the Italian [AWD](#) annotations in Figure 25.7. The reason for this is the dependency relation between the first [constituent](#) *church* and the fourth [constituent](#) *projects*, which leads to the smallest [AWD](#) between them. As we are using the [adjacency model](#) (i.e., considering only pairs of adjacent [constituents](#)), the smallest [AWD](#), 1, is annotated on the root node. We will discuss an alternative model that combines the [adjacency model](#) and the [dependency model](#) for [kCs](#) ( $k > 3$ ) in our discussion of future work (see Section 26.3.4).

While there are no valid full [parse trees](#), the righthand [parse tree](#) (shown in Fig-

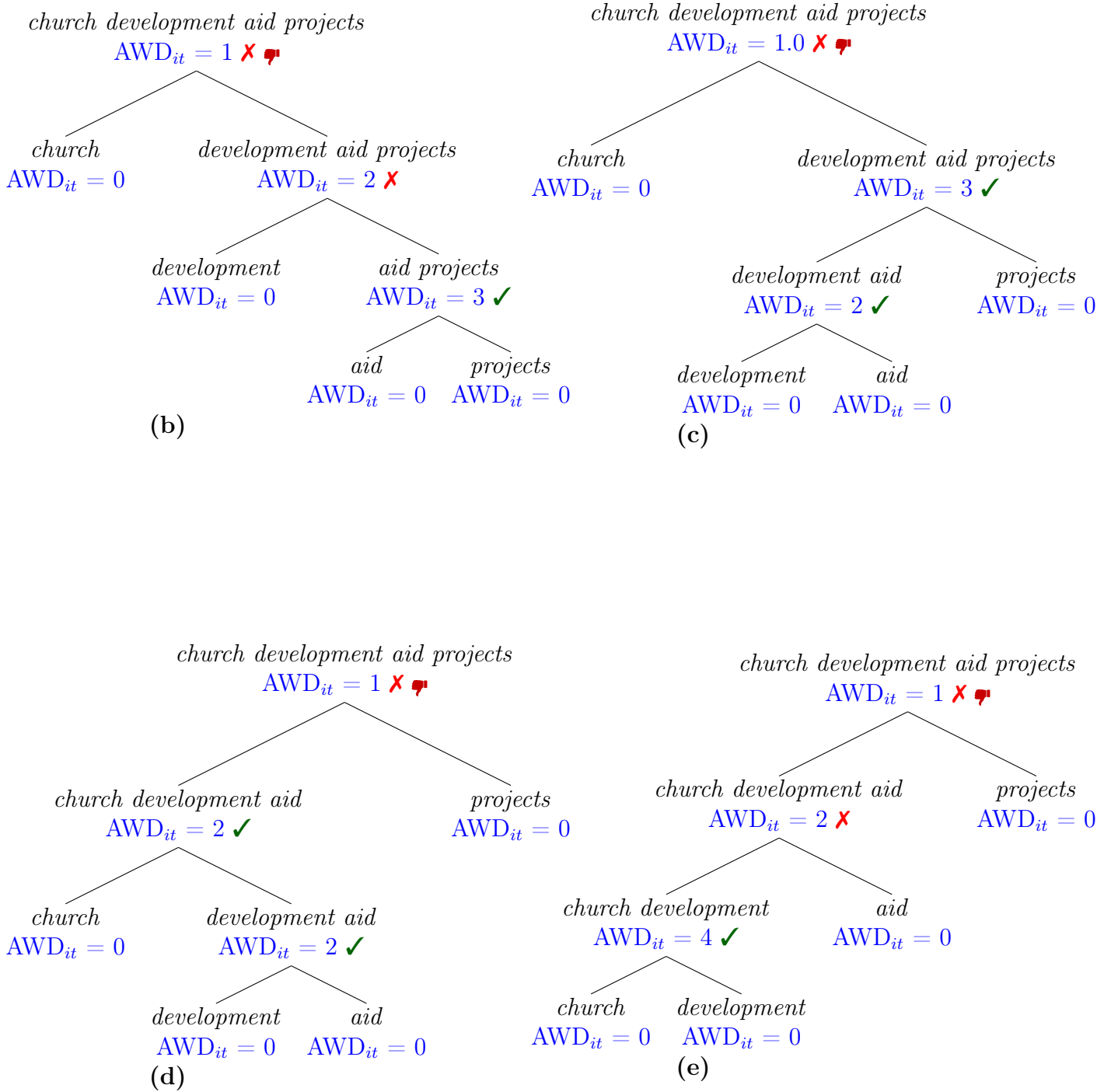


Figure 25.7.: Possible binary parse trees for *church development aid projects*

ure 25.7(c)) provides a valid **subtree** for the **constituent** sequence *development aid projects*. Neglecting this information leads to increased data sparsity of **phrasal equivalents** of **target constituents** in **parallel data**. Therefore, we developed the **Non-deterministic Subtree Accumulation Parsing (NSTAP)** that accumulates valid **subtrees** instead of valid full **parse trees**.

## Algorithm

---

### Algorithm 25.3 Non-deterministic Subtree Accumulation Parsing

---

**Target:** Compound  $\Psi$

```

1: Trees  $\leftarrow$  generate all possible full binary parse trees for  $\Psi$ 
2: STA  $\leftarrow$  []           {the STA collects all valid subtree instances}
3: for support language  $l_i \in L$  do
4:   for full parse tree  $ft \in \mathbf{Trees}$  do
5:     annotate all nodes  $N_i$  in  $ft$  with  $\mathbf{AWD}_{l_i}$ 
6:     for Node  $N_i$  in  $ft$  do
7:        $st_i \leftarrow \mathbf{subtree}(N_i)$            {create a subtree  $st_i$  of  $ft$  with  $N_i$  as root node}
8:       if  $\mathbf{valid}(st_i)$  then
9:         STA  $\leftarrow$  STA + [ $st_i$ ]           {if  $st_i$  is valid, it is added to the STA}
10:      end if
11:    end for
12:  end for
13: end for
14: for full parse tree  $ft \in \mathbf{Trees}$  do
15:    $\mathbf{treeScore}(ft)$ 
16: end for
17: return { $ft \in \mathbf{Trees} \mid ft$  has largest  $\mathbf{treeScore}$ }

```

---

Algorithm 25.3 shows the algorithm for **NSTAP**. After generating all possible **binary parse trees** for a **target compound**  $\Psi$  (line 1), the **SubTree Accumulation (STA)** (a list of valid **subtree** instances) is initialized (line 2). For all aligned **support languages**  $l_i \in L$ , we annotate all full **parse trees**  $ft$  with the **AWDs** (lines 2-5), as has been done for **NFTAP**. For each node  $N_i$  in a full **parse tree**  $ft$ , we generate the corresponding **subtree** having  $N_i$  as **root node** (lines 6-7). If this **subtree** is semantically valid (25.3.1), the **subtree** is added to the **STA** (lines 8-10).

Finally, all full **parse trees** are scored according to a  $\mathbf{treeScore}$  given in Formula 25.7, where  $\mathbf{freq}(st_i, \mathbf{STA})$  is the frequency of a **subtree**  $st_i$  in the **STA**,  $|L|$  is the number of **support languages** and  $\mathbf{Cat}_\Delta$  is the  $\Delta$ -th Catalan number (Formula 25.5) with  $\Delta$  being the difference in the number of **leaf nodes** between  $ft$  and  $st_i$ .

$$\begin{aligned}
treeScore(ft) &= \prod_{st_i \in ft} P(valid|st_i) \\
&= \prod_{st_i \in ft} \frac{freq(valid \cap st_i)}{freq(st_i)} \\
&= \prod_{st_i \in ft} \frac{freq(st_i, STA)}{|L| \cdot Cat_\Delta}
\end{aligned} \tag{25.7}$$

For example, we are given a subtree  $st_\alpha$  having **three** leaf nodes from a full parse tree  $ft_\beta$ , and  $ft_\beta$  has a total of **five** leaf nodes. The subtree  $st_\alpha$  is **four** times valid within a set of **nine** support languages. The factor for  $st_\alpha$  (in Formula 25.7) is given in Formula 25.8.

$$\begin{aligned}
P(valid|st_\alpha) &= \frac{freq(st_\alpha, STA)}{|L| \cdot Cat_\Delta} \\
&= \frac{4}{9 \cdot C_{5-3}} \\
&= \frac{4}{9 \cdot 2} = \frac{2}{9} \sim 0.222
\end{aligned} \tag{25.8}$$

In the last step, all full parse trees having the largest *treeScore* are returned (line 17). If there are more than one parse tree having the largest *treeScore*, NSTAP does not produce a final structure. Although the continual values of *treeScore* makes it less likely to have several parse trees with the same top-score, a non-deterministic output in NSTAP can be considered two-fold (i.e., predicting semantic indeterminacy and applying a subset of possible parse trees to other downstream parsing methods), as already discussed for DBUP and NFTAP.

Similarly to DBUP and NFTAP, NSTAP is also applicable token-based (i.e., accumulating the valid subtrees of all possible parse trees derived from the aligned sentences of a certain instance of the target compound  $\Psi$ ) and type-based (i.e., accumulating the valid subtrees of all possible parse trees derived from the aligned sentences of **all** instances of the target compound  $\Psi$ ).



### Example case

To illustrate the procedure of **NSTAP**, we revisit the **4NC** from our initial example, *church<sub>A</sub> development<sub>B</sub> aid<sub>C</sub> projects<sub>D</sub>* aligned to phrases of three **support languages**: Italian *progetti<sub>D</sub> ecclesiastici<sub>A</sub> di aiuti<sub>C</sub> allo sviluppo<sub>B</sub>* (lit: ‘projects ecclesiastical of aid to development’), German *kirchliche<sub>A</sub> Entwicklungshilfeprojekte<sub>BCD</sub>* ‘church {development aid projects}’ and French *projets<sub>D</sub> d’ aide<sub>C</sub> au développement<sub>B</sub> de l’ Église<sub>A</sub>* ‘projects of aid for development by the church’.

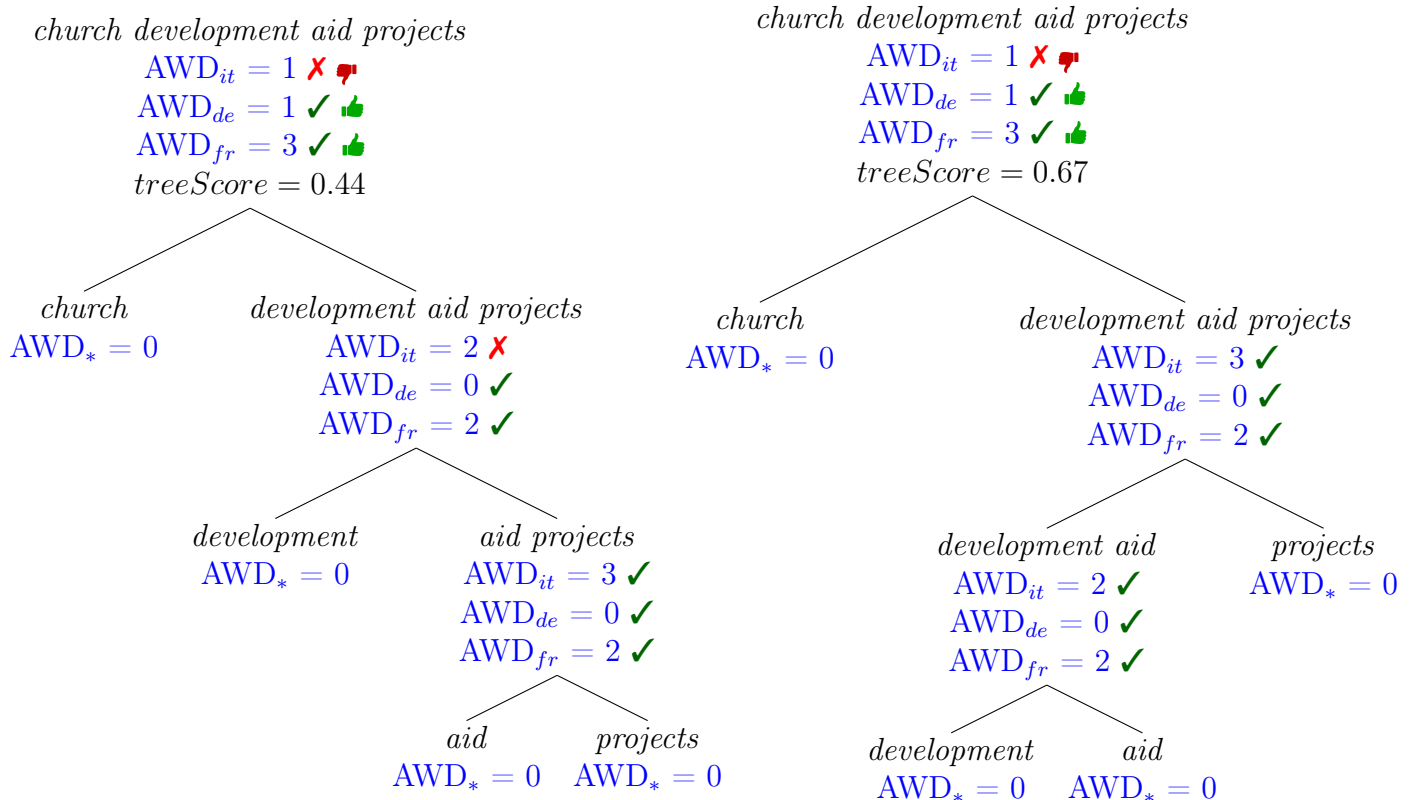


Figure 25.8.: Two possible **parse trees** for *church development aid projects* annotated with **AWDs** in Italian, German and French

When accumulating full **parse trees** using **NFTAP**, the two **parse trees** given in Figure 25.8 are both ranked on top in the **FTA**: both **parse trees** are twice valid and once invalid. Thus, it is not possible to determine a more plausible structure among them. Switching to the **subtree** accumulation in **NSTAP**, we can score both full **parse trees** according to the amount of contained valid **subtrees**. Due to the fact that the **subtree** spanning *development aid projects* is three times correct for the right **parse tree** in Figure 25.8 and only twice for the left **parse tree**, the right **parse tree** gets the higher

*treeScore* and is thus selected as the most plausible tree structure.

As final remark, it has to be said that both [parse trees](#) in Figure 25.8 are plausible, i.e., *development aid projects* can be considered as [semantically indeterminate](#). When using the Italian [phrasal equivalent](#), we see that the LEFT-branched structure for *development aid projects* is more prominent.

### 25.3.4. Experiments

#### Data

As for our pilot study (Chapter 24) and the experiments on [DBUP](#) (Section 25.2), we used the [CCR\(0\)](#) version of the [ENCD](#) as grounding source for the [kNCs](#) and their [equivalents](#) in 9 European [support languages](#). We extracted [3NCs](#) and [4NCs](#) from the database using a [PoS pattern](#) modelling a sequence of three to four adjacent nouns. The dataset contains 24,848 [3NC tokens](#) (16,565 [types](#)) and 1468 [4NC tokens](#) (1257 [types](#)).

#### Gold Standard Annotation

For evaluating the quality of [parsing 3NCs](#), we used the test samples developed for [DBUP](#) (Section 25.2.3). Besides the 278 samples that are labelled as either LEFT or RIGHT, there are 120 samples that have been classified as SEMIND (i.e., both LEFT- and RIGHT-branching, or [semantically indeterminate](#)).

Besides [3NCs](#), we also decided to evaluate our [compound parsers](#) on [4NCs](#). One reason for this is that there are more [semantically indeterminate 4NCs](#) than [3NCs](#) (as will be shown later). Another reason is that [4NCs](#) have five possible [binary parse trees](#), whereas [3NCs](#) have only two possible [parse trees](#) (as has been shown in Table 25.4). Therefore, we consider the [parsing of 4NCs](#) as more challenging.

For creating a test set of [4NCs](#) annotated with structure, we had to decide for a size which has a similar ratio of the total number of respective [kNC types](#) in the dataset as for the [3NCs](#), which was (278+120 samples / 16,565 [types](#)) roughly 2.4%. Since the same ratio applied to [4NCs](#) would roughly mean (2.4% of 1257 [types](#)) 30 samples, we decided to adjust this number upward to 50 samples for avoiding issues of data sparseness.

In the [4NC](#) annotation process, we adopted the guidelines of Vadas (2009) and used the following nine labels:

**1 . . . 5:** one of the five possible [4NC](#) structures, represented as [bracketing patterns](#)

**ERROR:** for falsely extracted **compounds**, e.g., incomplete **4NCs** or **PoS** errors as in *climate change target cannot*

**UNKNOWN( $i, \dots, j$ ):** the **4NC** cannot be disambiguated between the distinctive structures  $i, \dots, j$  within the one-sentence context

**FLAT:** for expressions showing no **internal structure**, e.g., named entities like *John A. Smith*

**SEMIND( $i, \dots, j$ ):** the **4NC** is **semantically indeterminate**, i.e., the structures  $i, \dots, j$  are semantically equivalent

In the final test set, we combine samples annotated with SEMIND and with a certain structure label (1 . . . 5), resulting in a set of 33 **4NC** samples. The remaining 17 samples were labelled as ERROR.

Bracketing pattern	Frequency									
	13	6	5	2	1					
A [B [C D]]		*					*	*	*	
A [[B C] D]				*				*	*	*
[A B] [C D]	*		*			*	*	*		
[A [B C]] D						*				*
[[A B] C] D	*				*	*				

Table 25.6.: Frequency distribution of **bracketing patterns** in the **4NC** test set

Table 25.6 shows the frequency distribution of the 33 samples. The columns 2 to 11 show how often a certain structure combination has been labelled. One interesting observation is that, in analogy to the majority class LEFT for **3NCs** (i.e., the **LEFT class baseline**), the majority **bracketing pattern** for **4NCs** represents a combination of structures where the two leftmost nouns (A and B) form a **constituent**.

## Evaluation Measures

As described above, the output of **NTAP** is a ranked list of **parse trees**. Inspired by **Information Retrieval (IR)** models, we treat **NTAP** as a kind of **structure retrieval** and measure how well **parse tree** ranking fits to the set of gold trees.

As first measure, we adapted the R-Precision score (Buckley and Voorhees, 2000) as given in Formula 25.9.

$$R\text{-Prec}(kNC) = \frac{|\text{top-}R(\text{sys trees}) \cap \text{gold trees}|}{|\text{top-}R(\text{sys trees})|} \quad (25.9)$$

where  $R$  is the number of gold trees and  $\text{top-}R(\text{sys trees})$  refers to the set of the  $R$  highest-ranked system [parse trees](#) and *gold trees* refers to the set of  $R$  gold trees.

If there are several [parse trees](#) having the same system rank, we choose a random order. If there are less than  $R$  [parse trees](#) predicted by the system (e.g., a [semantically indeterminate kNC](#) has evidence for only one structure), the ranking is randomly complemented to  $R$  [parse trees](#). Observing that this random process leads to unstable numbers due to the small gold standard size, we applied the random process 1000 times and took the average of the resulting score.

The mean  $R$ -Precision MRP takes the macro average of the  $R$ -Precision scores, as given in Formula 25.10

$$\text{MRP}(\Omega) = \frac{\sum_{\Psi \in \Omega} R\text{-Prec}(\Psi)}{|\Omega|} \quad (25.10)$$

where  $\Omega$  is the set of all [target compounds](#)  $\Psi$  (i.e., [3NCs](#) and [4NCs](#)) in our test set.

As further [IR](#)-inspired measures, we used precision at  $t$  ( $P@t$ ) and recall at  $t$  ( $R@t$ ) as given in Formula 25.11 and 25.12.

$$P@t = \frac{|\text{top-}t(\text{sys trees}) \cap \text{gold trees}|}{|\text{top-}t(\text{sys trees})|} \quad (25.11)$$

$$R@t = \frac{|\text{top-}t(\text{sys trees}) \cap \text{gold trees}|}{|\text{gold trees}|} \quad (25.12)$$

We present the macro average for  $P@t$  as  $MP@t$  and for  $R@t$  as  $MR@t$ . Macro  $F_1$  at  $t$  is the harmonic mean of  $MP@t$  and  $MR@t$ . Since [semantically indeterminate kNCs](#) have about two gold [parse trees](#) in our test set, we evaluate the systems for  $1 \leq t \leq 2$ .

## Methods in Comparison

We compare [NFTAP](#) and [NSTAP](#) against [DBUP](#). Since [DBUP](#) uses the majority vote as deterministic output, it cannot detect [semantic indeterminacy](#). Thus, we additionally

modified **DBUP** such that it produces a frequency-ranked output of the **parse trees** (still providing at most one **parse tree** per **support language**). We call this method **DBUP<sub>rank</sub>**.

While we compared **DBUP** against the  $\chi^2$ -based **compound parser** in Section 25.2.3, we did not include it in the current experiments. One reason for this is that the  $\chi^2$ -based method is defined for the **binary LEFT/RIGHT** classification of **3NCs**. Using  $\chi^2$  as an **Association Measure (AM)** between word sequences (as is done with the **AWD** metric) provides issues of data sparsity and runtime complexity. Another reason is that it is not clear how we can adapt  $\chi^2$  in order to get a ranking output of several **parse trees** (for **identifying semantically indeterminate 4NCs**). Finally, **DBUP** already outperforms the  $\chi^2$ -based method in **parsing** accuracy. For the case that **NTAP** outperforms **DBUP** in accuracy, we could infer that it would also outperform  $\chi^2$ .

As baselines, we used the random baseline **CHANCE**, which creates an arbitrary **parse tree** ranking, and the frequency baseline **FREQ**, which creates a tree ranking according to the **bracketing pattern** frequencies in the test set shown in Table 25.6, i.e., the **parse tree** conforming with the most frequent **bracketing pattern** (e.g., [A B] [C D]) is ranked highest. The **FREQ** baseline corresponds to the **LEFT class baseline** for **3NCs** which are not **semantically indeterminate**.

While the **4NC** structure votes of both independent annotators (i.e., **SEMIND** and structure labels (1 . . .5)) have been combined rather than intersected (as has been described above in the Gold Standard Annotation), the author of this thesis provided an additional annotation layer of the **4NC** test set, representing the upper bound, **UPPER**. Since we used the consensus of **LEFT/RIGHT** decisions (**IAA** rate of 90.3%) for our **3NC** gold standard (as has been described in Section 25.2.3) and added **3NCs** annotated as **semantically indeterminate**, there is no need for an additional **3NC** annotation layer serving as upper bound.

Since the experiments for **DBUP** (25.2.3) showed that the **type-based** outperforms **token-based compound parsing** for **kNCs** in the **ENCD**, we decided to evaluate all models on **types**.

## Results

Table 25.7 shows the results of the mean *R*-Precision (MRP) on the test set of **3NCs** and **4NCs**. All methods for **cross-lingual compound parsing** outperform the two baselines **FREQ** and **CHANCE**. Moreover, **NFTAP** and **NSTAP** outperform **DBUP** and **DBUP<sub>rank</sub>**, but the differences are small (average difference: 1.55%).

Considering the ratios of **semantic indeterminacy**, we can see that there are more

System	MRP
NFTAP	93.7%
NSTAP	<b>94.0%</b>
DBUP	92.6%
DBUP <sub>rank</sub>	92.0%
FREQ	84.6%
CHANCE	62.5%

Table 25.7.: Parsing results in MRP for 3NCs and 4NCs

semantically indeterminate 4NCs ( $\frac{18}{33} = 54.5\%$ ) than 3NCs ( $\frac{120}{398} = 30.2\%$ ). Since a benefit of NTAP is to detect semantic indeterminacy, we expect to see larger differences between the deterministic DBUP and the NTAP methods, when evaluating on 4NCs separately.











System	MRP	MP@1	MR@1	MF1@1	MP@2	MR@2	MF1@2
NFTAP	<b>70.0%</b>	<b>72.7%</b>	<b>47.5%</b>	<b>57.5%</b>	60.6%	74.2%	66.7%
NSTAP	69.5%	69.7%	44.4 %	54.2%	<b>63.6%</b>	<b>78.8%</b>	<b>70.4%</b>
DBUP	54.5% 	69.7%	44.4%	54.2%	47.0% 	59.1% 	52.4% 
DBUP <sub>rank</sub>	62.9% 	69.7%	44.4%	54.2%	54.5% 	66.7% 	60.0 % 
UPPER	86.0%	96.7%	67.2%	79.3%	70.0%	87.8%	77.9%
FREQ	60.1%	63.6%	38.4%	47.9%	56.1%	65.2%	60.3%
CHANCE	32.0%	39.4%	23.7%	29.6%	33.3 %	42.4 %	37.3%

Table 25.8.: Parsing results for 4NCs

Table 25.8 shows the results on cross-lingual compound parsing of 4NCs, where  means significantly outperformed by both NTAP methods;  means significantly outperformed by NFTAP or NSTAP. For the mean  $R$ -Precision, MRP, NFTAP and NSTAP significantly<sup>6</sup> outperform DBUP and DBUP<sub>rank</sub>. Precision and Recall at 1 are similar for all methods for cross-lingual parsing, i.e., the top position of the systems' rankings hardly differ (slight advantage for NFTAP). For Precision and Recall at 2, the NTAP methods significantly outperform DBUP and DBUP<sub>rank</sub> (slight advantage for NSTAP).

A final observation is that NSTAP performs similarly to NFTAP. This shows that the benefit of NSTAP (i.e., exploiting valid subtrees for support languages that do not provide any valid full tree, like *Italian*, as shown in Figure 25.7) does not come into effect with our parallel corpus comprising nine support languages. We expect NSTAP to clearly outperform NFTAP with parallel corpora providing fewer support languages.

<sup>6</sup>Approximate randomization test (Yeh, 2000),  $p < 5\%$

# 26. Bottom Line of the Compound Parsing

This chapter constitutes the bottom line of the [compound parsing](#) Part E. In Section 26.1, all previous chapters on [compound parsing](#) are summarized. Then, in Section 26.2 we draw some conclusions and discuss the research questions posed in Section 22.2. Finally, we give an outlook on possible future work in Section 26.3.

## 26.1. Summary

In [Chapter 22](#), we introduced the [compound parsing](#) Part E, i.e., we discussed some motivations such as the importance of [compound parsing](#) (22.1.1) and presented our **guiding principle** (22.1.2) that describes the correlation between spatial proximity and [semantic association](#) as initially suggested by Behaghel (1909). Our **contributions** in [compound parsing](#) (22.2) are the usage of spatial proximity as a measure for [semantic association](#) (22.2.1), the [cross-lingual](#) perspective we take that allows for [token-based compound parsing](#) (22.2.2), a well-defined metric for measuring spatial proximity across languages (22.2.3) and the ability to automatically detect [semantic indeterminacy](#) using [cross-lingual](#) evidence (22.2.4).

In [Chapter 23](#), we outlined previous and related work on [compound parsing](#). Firstly, we described two basic approaches that are often used in different ways within the task of [parsing ternary compounds](#) (23.1), i.e., the [adjacency model](#) (23.1.1), the [dependency model](#) (23.1.2) and a hybrid approach (23.1.3). Then, we discussed statistical [Association Measures](#) that have been used commonly for [compound parsing](#) (23.2). Previous work on [parsing noun compounds](#) was presented in Section 23.3. A more general [target](#) class are [base NPs](#), whose previous [parsers](#) were described in Section 23.4. Finally, we outlined further related work on the structural disambiguation of other linguistic expressions and cases of syntactic ambiguity (e.g., sentence [parsing](#) or [PP-attachment](#) ambiguity) using [cross-lingual](#) support (23.5).

A **pilot study** on **cross-lingual compound parsing** is presented in Chapter 24. First, we presented the main subject of our pilot study, the **Aligned Phrase Patterns (APPs)** (24.1), i.e., their function (24.1.1), the manual definition of APPs (24.1.2) and the way how the structure class (i.e., LEFT or RIGHT) correlates with the APP (24.1.3). Based on the APPs, we developed a **cross-lingual compound parser**, the **Aligned Phrase Pattern Parsing (APPP)** (24.2). The regular APPP does not work for cases of **constituent swapping**. Thus, we revised the method to  $\text{APPP}_{WA}$  by adding the support of **word alignment** information (24.3). In an experiment on 76 **three-Noun Compounds** extracted from the ENCD, we observed that our first **cross-lingual compound parser** shows a solid performance and significantly outperforms the **LEFT class baseline** (24.4).

As discussed in Section 24.4.6, the APPP method has several crucial limitations such as the coverage, i.e., there are many **compounds** whose **aligned phrases** cannot be mapped on an APP with a known structure class. Thus, in **Chapter 25** we abstracted away from APPs to a **cross-lingual metric for measuring spatial proximity** as an approximation for **semantic association**, the **aligned word distance (AWD)** (25.1). The AWD metric is the basis for several more advanced **compound parsing** methods. The first approach was the **deterministic bottom-up parsing (DBUP)** (25.2). DBUP iteratively merges pairs of adjacent **target constituents** with the smallest AWD for a given **support language**. The algorithm works bottom-up starting with **atomic target constituents** (25.2.1). In an experiment on **parsing 278 three-Noun Compounds**, we observed that DBUP is clearly superior to APPP in coverage (Table 25.1). (25.2.3)

The main limitation of DBUP is the exploitation of partial **parsing** results provided by a **support language**. For example, the **4NC book price fixing schemes** aligned to the Danish **phrasal equivalent** *fastprisordningerne for bøger* ‘price fixing schemes for books’ cannot be fully **parsed**. However, the Danish **equivalent** provides the knowledge that the **target constituent** *books* is adjoined to *price fixing schemes* at the **root node**, as illustrated in Figure 25.3. In order to solve this issue, we developed two methods for **Non-deterministic Tree Accumulation Parsing (NTAP)** (25.3). Firstly, we described a principle of semantically valid **parse trees** (25.3.1), which is used for the subsequent NTAP methods. As first NTAP method, we presented an approach that is based on full **parse trees**, the **non-deterministic full tree accumulation parsing (NFTAP)** (25.3.2). Here, all possible **binary parse trees** for a given **target compound** are generated. For a **support language**  $l$ , all nodes in these trees are annotated with the  $\text{AWD}_l$  between the **constituents** related to their daughter nodes. According to the principle of semantically valid **parse trees** (that is based on bottom-up monotonically



increasing AWD annotations), the binary parse trees are validated and only valid trees are returned. After applying NFTAP to all support languages, all valid trees are accumulated and the most frequent parse trees are selected as structure prediction. We illustrated the performance of NFTAP for deterministic parsing and semantic indeterminacy detection with two examples.

A limitation of NFTAP is that it only accepts fully valid parse trees and neglects the fact that there are invalid parse trees that provide valid subtrees, as has been exemplified for the 4NC *church<sub>A</sub> development<sub>B</sub> aid<sub>C</sub> projects<sub>D</sub>* being aligned to the Italian phrasal equivalent *progetti<sub>D</sub> ecclesiastici<sub>A</sub> di aiuti<sub>C</sub> allo sviluppo<sub>B</sub>*. Due to the smallest AWD<sub>it</sub> between *church<sub>A</sub>* and *projects<sub>D</sub>*, the root node's AWD annotation always produced an invalid parse tree. A solution for this issue is proposed with the **Non-deterministic Subtree Accumulation Parsing (NSTAP)** (25.3.3). Instead of accumulating valid full parse trees, in NSTAP we accumulated all valid subtrees across all support languages. All full parse trees  $ft_i$  are finally ranked according to a *treeScore*, which reflects the product of subtree validity probabilities for all included subtrees of  $ft_i$ , as shown in Formula 25.7.

Finally, we presented some experiments on parsing 398 3NCs and 33 4NCs using the NTAP methods in Section 25.3.4. In order to take into account semantic indeterminacy when parsing compounds and inspired by IR, we treated the parsing task as a kind of **structure retrieval** and used the mean *R*-Precision (Formula 25.10) for evaluation. We compared the NTAP methods against DBUP, two baselines and an upper bound. The results proved that the NTAP methods are superior to DBUP. In particular for the set of 4NCs, which showed more cases of semantic indeterminacy than the 3NCs did, the advantage of NTAP led to a significantly better structure retrieval performance (see Tables 25.7 and 25.8).

And at the final end, this Chapter 26 summarizes (26.1) and concludes (26.2) the compound parsing part E and gives an outlook to future work (26.3).

## 26.2. Conclusion

In this section, we aim to answer the research questions posed in Section 22.2.

**RQ\_2-A:** What sources of indirect supervision can we use for compound parsing?

⇒ In this compound parsing part, we make use of cross-lingual supervision in terms of word distance between constituent equivalents in cross-lingually aligned sentences.

This kind of [cross-lingual supervision](#) is based on our guiding principle inspired by the First Law of Behaghel (1909), as described in Section [22.1.2](#).

**RQ\_2-A-iii:** What potential does our guiding principle ([22.1.2](#)) have for [cross-lingual compound parsing](#)?

⇒ In general, our guiding principle (i.e., the [cross-lingual](#) application of the First Law of Behaghel (1909)) is a very promising feature for [compound parsing](#). We approximate [semantic association](#) by exploiting spatial proximity in [cross-lingually](#) aligned sentences. In our pilot study (Chapter [24](#)), we exemplified the usage of our guiding principle by defining [Aligned Phrase Patterns \(APPs\)](#) having a complex and a simplex unit separated by [function words](#), and used the [APPs](#) in a [parser \(APPP\)](#). As discussed in Section [24.1.2](#), there is one [APP](#) which we disregarded in [APPP](#) because it violates our guiding principle (at least with respect to [atomic constituents](#)):  $SN_1 \quad FC \quad SN_2 \quad ADJ$  (i.e., a simplex noun followed by a functional context and a simplex noun with a postnominal adjective). This [APP](#) often occurs in Romance paraphrases for English [3NCs](#). While in most cases, the postnominal adjective refers to the closest noun  $SN_2$  (pointing to a LEFT-branched [3NC](#)), in a substantial amount of cases, the adjective refers to the entire [complex nominal](#)  $SN_1 \quad FC \quad SN_2$  or to the [head](#)  $SN_1$  (pointing to a RIGHT-branched [3NC](#)). An alternative [APP](#) for RIGHT-branched predictions that is consistent with our guiding principle (with respect to [atomic constituents](#)) is  $SN \quad ADJ \quad FC \quad SN$ , the fifth [APP](#), presented in Table [24.3](#).

On the other hand, as discussed in Section [25.1](#), while a difference in [word distance](#) proved to be a precise indicator for the [internal structure](#) of [compounds](#), an equal distance between pairs of [target constituents](#) does not necessarily mean an equivalent [semantic association](#). For example, the LEFT-branched [3NC](#) *risk<sub>A</sub> management<sub>B</sub> decision<sub>C</sub>* is aligned to the Swedish [ternary closed compound](#) *riskhanteringsbeslut<sub>ABC</sub>* and to the Portuguese phrase *decisão<sub>C</sub> de gestão<sub>B</sub> dos riscos<sub>A</sub>*. In both languages,  $AWD(risk, management)$  is equal to  $AWD(management, decision)$ .

**RQ\_2-A-iv:** Does the [token](#)-based approach, provided by the [cross-lingual](#) perspective, lead to a better [parsing](#) performance?

⇒ In general, structurally ambiguous [compounds](#) can be translated to expressive paraphrases (e.g., by revealing the structure by different spatial distances of the [equiva-](#)

lents of the **target constituents**). A human translator selects an expressive **phrasal equivalent** that reflects the **internal structure** of the intended meaning of the **target compound** in the **underlying context**. Thus, the **cross-lingual** perspective of our **compound parsing** methods, which enables a **token-based** process, has the potential of yielding better performance. Actually, the experiments described in Section 25.2.3 showed that  $APPP_{token}$  outperforms  $APPP_{type}$  in **parsing** accuracy by 0.6%.

However, there are several factors that hide (isolated or in combination) the advantage of **token-based parsing** in our experiments. Firstly, most **3NC types** in the **ENCD** have a dominant intended meaning which is reflected in the **aligned phrases** of all **3NC** instances (i.e., there is hardly any context-dependent ambiguity). Secondly, when counting coverage to the **parsing** quality, a **type-based** approach mitigates the issue of data sparsity with respect to expressive **APPs**. Finally, the determination of **aligned phrases** is based on statistical **word alignment**, which happens to produce noisy results, that could lead to non-expressive **APPs** or even to false structure predictions. The increased number of structure class votes (when switching from a **tokens** to **types**) mitigates the impact of noise due to **word alignment errors**.

In future work (see Section 26.3.6), we will create an evaluation setup that excludes the factors described above, i.e., we will select a test set containing exclusively structurally ambiguous **3NCs**, and we will manually correct all **word alignments** which are relevant for the **3NC** samples. With this setup, we expect to see a better performance for **token-based compound parsing**.

**RQ\_2-A-v:** How useful is the proposed **AWD** metric?

⇒ The **AWD** metric turned out to be a precise measurement for estimating the **cross-lingual** distance of two **target constituents**. The flexibility of **AWD** allows us to measure the **cross-lingual** distance between both **atomic** and complex **constituents**. Thus, the **AWD** is a metric that can be used for **parsing compounds** of any **size** (in terms of **atomic constituents**). Most traditional statistical **Association Measures (AMs)** (some of which are used as metric for **compound parsing** in previous work) are defined for pairs of unigrams. The computation of statistical association between  $N$ grams ( $N > 1$ ) is costly, time-consuming and often suffers from data sparsity.

However, there is one limitation of the AWD metric which is based on the simplification of treating all words between two underlying target constituents equally. It is plausible that content words represent a stronger separator than function words. Moreover, even different function words can indicate a different semantic association. For example, the LEFT-branched 3NC *labour market access* being aligned to the Italian phrasal equivalent *accesso al mercato del lavoro* (lit: ‘access to the market of labour’) indicates different semantic relations (and thus different degrees of semantic association) between the constituents due to the different Italian prepositions *al* and *del* (Girju, 2007). In the current way, the AWD metric does not treat *al* and *del* differently. The development of a more elaborated AWD measurement will be addressed in future work (see Section 26.3.2).

**RQ\_2-A-vi:** How competitive is cross-lingual compound parsing compared to other knowledge-lean parsers?

⇒ In the experiments described in Section 25.2.3, we compared DBUP and APPP against a statistical adjacency model which uses the Chi Squared ( $\chi^2$ ) measurement (Nakov and Hearst, 2005). While our cross-lingual methods are inferior in coverage, the parsing accuracy of our methods are fairly competitive to the  $\chi^2$ -based approach. When using the back-off models of APPP and DBUP (backing off to the  $\chi^2$  approach when the 3NC cannot be processed cross-lingually),  $APPP_{type} \rightarrow \chi^2$  is less than one percentage point worse than the  $\chi^2$ -based approach and  $DBUP_{type} \rightarrow \chi^2$  is about six percentage points better (see Table 25.2).

**RQ\_2-A-vii:** How precise is the cross-lingual detection of semantic indeterminacy?

⇒ Two of the proposed cross-lingual compound parsing methods, APPP and DBUP, are deterministic when applied to a single support language. However, when accumulating the parse trees derived from several support languages, these methods can be used for detecting cases of semantic indeterminacy: if there are several equally (or similarly) frequent parse trees, the compound can be considered as semantically indeterminate with respect to these structures.

The highest potential of detecting cases of semantic indeterminacy are the Non-deterministic Tree Accumulation Parsing (NTAP) methods, where we can predict several parse trees from each support language. We illustrated the potential of semantic indeterminacy detection with the example of *farm income stabilisation*

*instrument*, which is [semantically indeterminate](#) with respect to two [parse trees](#), shown in Figure 25.6. All [support languages](#) provide spatial evidence for *farm* and *income* having the strongest [semantic association](#), and for *income+stabilisation* and *stabilisation+instrument* having the same/comparable [semantic association](#). Taking into account cases of [semantic indeterminacy](#), in Section 25.3.4, we evaluated the [compound parsing](#) task as a kind of structure retrieval using measurements inspired by IR. The experiments’ results showed that the NTAP methods achieve a solid performance in [parsing semantically indeterminate kNCs](#), i.e., in retrieving all correct [parse trees](#).

## 26.3. Future Work

In this section, we discuss some limitations of the presented [cross-lingual compound parsing](#) methods and suggest ways to overcome these. Moreover, we propose some possible future research directions.

### 26.3.1. Parsing with Bootstrapped Aligned Phrase Pattern Set

In the [Aligned Phrase Pattern Parsing \(APPP\)](#), presented in our pilot study in Chapter 24, we used a set of six predefined [Aligned Phrase Patterns \(APPs\)](#) (24.1). Depending on the granularity of the USP format (see Appendix A), we could introduce new [APPs](#) for [parsing 3NCs](#) which are not considered for now, for example ADJC SN, i.e., a complex adjective followed by a simple noun, pointing to a LEFT-branched structure as in *steuerpolitische<sub>AB</sub> Entwicklung<sub>C</sub>* ‘tax policy trend’ (lit: ‘tax-political development’). Moreover, when switching from [3NCs](#) to [4NCs](#) or [kNCs](#), there are many more possible structures and related [APPs](#). In these cases, a hand-crafted set of [APPs](#) associated with their structure classes is costly and time-consuming, as discussed in Section 24.4.6.

A possible solution for this is to automatically derive [APPs](#) for a certain structure by using [bootstrapping](#). This idea is inspired by the [lexicon bootstrappers](#) proposed in previous work, such as the one-word bootstrapper BASILISK (Thelen and Riloff, 2002) or the coordination-based [MWE bootstrapper](#) BASILISK-C (Ziering et al., 2013a). A possible pseudo-algorithm of [bootstrapping APPs](#) is given in Algorithm 26.1. The method is initialized with a seed list of [APPs](#) for a given [compound size](#)  $k$  and a structure class  $\Sigma$ . Moreover, a bidirectional mapping resource that aligns [kNCs](#) onto all observed [APPs](#) and vice versa (e.g., the [ENCD](#)) is used. The method is repeated for  $m$  iterations (line

1). In the first step of the iteration, all **kNCs** being aligned to all known **APPs** are collected (line 2). These **kNCs** are scored according to the  $RlogF$  score initially proposed by Thelen and Riloff (2002), given in Formula 26.1, where  $F_j$  is the number of learned **APPs** that are aligned to the  $kNC_j$ , and  $N_j$  is the total number of **APPs** aligned to  $kNC_j$  (line 3).

$$RlogF(kNC_j) = \frac{F_j}{N_j} \cdot \log_2(F_j) \quad (26.1)$$

---

**Algorithm 26.1** Bootstrapping of **APPs** for a given **compound size** and structure  $\Sigma$

**APPs**: a seed-initialized lexicon of **APPs** for a **compound size** and structure  $\Sigma$

**Mapping**: a resource that maps **kNCs** onto all observed **APPs** and vice versa

```

1: for int  $i = 0; i < m; i++$  do
2:   kNCs  $\leftarrow$  kNCs(APPs)
3:   score(kNCs)
4:   kNCs  $\leftarrow$  return-top- $l$ (kNCs,  $20 + i$ )
5:   new APPs  $\leftarrow$  APPsMappedOnto(kNCs) - APPs
6:   score(new APPs)
7:   APPs  $\leftarrow$  APPs  $\cup$  return-top- $l$ (new APPs, 5)
8: end for
9: return APPs

```

---

In the next step, the top- $l$ <sup>1</sup> ranked **kNCs** are returned (line 4). New **APPs** are determined that are aligned to the collected **kNCs** but that are not already included in the **APP** collection (line 5). These new **APPs** are scored according to the  $AvgLog$  score proposed by Thelen and Riloff (2002), which is given in Formula 26.2, where  $K_n$  is the number of **kNCs** that are aligned to  $APP_j$  and  $F_o$  is the number of learned **APPs** being aligned to  $kNC_o$  (line 6).

$$AvgLog(APP_j) = \frac{\sum_{o=1}^{K_n} \log_2(F_o + 1)}{K_n} \quad (26.2)$$

The top-5 ranked **APPs** are added to the the **APP** list (line 7). Finally, after  $m$  iterations, the expanded list of **APPs** is returned (line 9).

While this method and its parameters are designed for the task of **semantic lexicon bootstrapping** (where **words** correlated with **APPs** and lexico-syntactic patterns corre-

---

<sup>1</sup>Thelen and Riloff (2002) used  $l = 20 + i$  as number of collected items, which increases with the number of iterations  $i$ .

late with **kNCs**), this approach has to be adapted to the task of **bootstrapping APPs**. There are some differences in the respective goals of **bootstrapping**, for example, there are fewer **APPs** and **kNCs** than **words** and lexico-syntactic patterns. Thus, the top-1 parameters have to be much smaller.

As an example, we bootstrap **APPs** for **4NCs** with the structure [A B] [C D]. As initialized seed, we use the **APP** *CN FC CN*. A possible **kNC** aligned to this seed **APP** is *energy efficiency action plan*. A potential **APP** being frequently aligned is **SN FC SN FC SN ADJ** as in the French *plan d'action pour l'efficacité énergétique*). While this **APP** is still structurally ambiguous, as discussed in Section 24.1.2, and it can be used to reduce the search space for pattern-based **compound parsing** (e.g., the more collected **APPs** for a structure  $\Sigma$  are aligned to a **target compound**  $\Psi$ , the more likely  $\Psi$  has the structure  $\Sigma$ ).

### 26.3.2. Weighted Aligned Word Distance

The proposed **AWD** metric works too simple, because it treats all the same, i.e., all **words** get the same weight. However, it is plausible that **content words** represent a stronger separator (and thus should get a higher weight) than **function words**. Furthermore, even different **function words** (i.e., parts of a functional context **FC**) should get different weights. While equal **AWDs** are fine for symmetric **complex nominals** such as the French *résultats de analyse de marché*, for asymmetric cases with different prepositions such as the French *personnes en situation de pauvreté* (*people in poverty*) the **AWD** metric should give the less frequent preposition *en* a higher weight than to the highly frequent preposition *de*. Another example is the **4NC** *aviation safety improvement strategy* aligned to the German **phrasal equivalent** *Strategie zur Erhöhung der Flugsicherheit* (lit: ‘strategy {for the} increase {of the} {aviation safety}’). Here, the conflated preposition *zur* should get higher weight than the genitive article *der*.

As a possible extension, we propose the **weighted aligned word distance** ( $w$ **AWD**), i.e., the regular **AWD** with a weighting function  $w$  applied to each **word** within the minimum context between the **equivalents** of two **target constituents**. For getting this minimum context, we define the function  $cAWP(c_i, c_j)$ , which determines the closest **Aligned Word pair** of the **target constituents**  $c_i$  and  $c_j$ , as shown in Formula 26.3, where the first aligned **word** in the pair is located first in the aligned sentence. If there are several aligned **word** pairs having the minimum distance,  $cAWP$  takes the first by default.

$$cAWP(c_i, c_j) = \arg \min_{x \in AWS(c_i), y \in AWS(c_j)} |pos(x) - pos(y)| \quad (26.3)$$

The final formula for  $wAWD$  is shown in Formula 26.4, where  $\Delta_{pos}(cAWP(c_i, c_j))$  returns the size of the minimum context between the **equivalents** of  $c_i$  and  $c_j$ , and  $w(\alpha_k)$  represents a weighting function applied to the  $k$ -th word in an aligned sentence.

$$wAWD(c_i, c_j) = \begin{cases} AWD(c_i, c_j) & \text{if } \Delta_{pos}(cAWP(c_i, c_j)) < 2 \\ 1 + \sum_{k=cAWP(c_i, c_j)[1] - 1}^{cAWP(c_i, c_j)[0] + 1} w(\alpha_k) & \text{else} \end{cases} \quad (26.4)$$

There are many possible weighting functions for  $wAWD$ . A first simple weighting function could distinguish between **content word** and **function word**, giving **content words** a doubled weight, as shown in Formula 26.5.

$$w(\alpha) = \begin{cases} 2 & \text{if } \textit{contentword}(\alpha) \\ 1 & \text{else} \end{cases} \quad (26.5)$$

Another possible weighting function is based on the observation that frequent French prepositions (e.g., *de*) represent a stronger **semantic association** than infrequent prepositions (e.g., *en*) do, i.e., the “distance” between two nouns linked by *en* should be considered larger than the distance between two nouns linked by *de*. The frequency-based weighting function is given in Formula 26.6.

$$w(\alpha) = \frac{1}{freq(\alpha)} \quad (26.6)$$

While the frequency-based weighting function works in many cases, there are some strong collocations including infrequent prepositions that are modified by nouns separated by more frequent prepositions. For example, the **3NC** *cable television packages* being aligned to the Portuguese **phrasal equivalent** *pacotes de televisão por cabo* (lit: ‘packages **of** television **for** cable’), where *de* is the more frequent Portuguese preposition than *por*, but the correct structure is  $[[cable\ television] packages]$ , i.e., *por* shows a stronger linkage than *de* in this context.



Besides refining the frequency-based weighting function, future work also includes finding further weighting functions and combining them with the proposed functions above.

As final remark, there is no different between  $w$ AWD and the regular AWD if the aligned words are identical (i.e.,  $AWD = 0$ , e.g., in the case of a common aligned **closed compound**) or adjacent (i.e.,  $AWD = 1$ ).

### 26.3.3. Monolingual Word Distance Metric for Compound Parsing

Nakov and Hearst (2005) used monolingual corpora for finding instances of paraphrases revealing a certain **bracketing**, e.g., *cells<sub>C</sub> from the brain<sub>A</sub> stem<sub>B</sub>* pointing to a LEFT-branched  $[[\textit{brain stem}] \textit{cell}]$ . In our pilot study in Chapter 24, we adopted this idea and developed APPs revealing a certain **bracketing** from a **cross-lingual** perspective. In Chapter 25, we presented methods relying on the AWD metric, within the **cross-lingual** perspective.

An interesting part of future work on **compound parsing** is to apply our guiding principle, i.e., the relation between spatial proximity and **semantic association** (22.1.2) to monolingual corpora. Therefore, we have to adapt and apply the AWD metric to instances of all **target constituents** within a suitable monolingual context  $\Lambda$  (e.g., a sentence). The modified AWD metric is given in Formula 26.7, where  $pos_{\Lambda}(\alpha)$  refers to the position of a **target constituent** instance  $\alpha$  within a given context  $\Lambda$ .

$$WD_{\Lambda}(c_i, c_j) = |pos_{\Lambda}(c_i) - pos_{\Lambda}(c_j)| \quad (26.7)$$

In the case that there are multiple occurrences of a **target constituent** instance, the one with the minimum **word distance** (WD) can be chosen, by default. For example, for **parsing** the 3NC *human rights violations*, the news<sup>2</sup> title *Report on the Violations of Human Rights in the Conflict Zones* can provide the correct structure:  $WD(\textit{human}, \textit{rights}) = 1$ ,  $WD(\textit{human}, \textit{violations}) = 2$  and  $WD(\textit{rights}, \textit{violations}) = 3$ . Having the smallest WD between *human* and *rights*, this context provide evidence for a LEFT-branched 3NC. Accumulating the results of various contexts leads to the final **parsing** result. It is subject of future work to find the optimal trade-off between contextual proximity (for ensuring that the **target constituent** instances in the context are still related) and data sparseness

---

<sup>2</sup>[osce.org/odhr/27773](http://osce.org/odhr/27773)

(for finding enough contexts including all [target constituent](#) instances). Unfortunately, in this monolingual setup, the [compound parsing](#) can only be done [type-based](#), disregarding context-dependent structural ambiguity.

### 26.3.4. Hybrid Adjacency-Dependency Model

As discussed in Section 25.3.3, counting [subtrees](#) in [Non-deterministic Subtree Accumulation Parsing](#) (NSTAP) can be useful for [4NCs](#) being aligned to phrases in which [word](#) positions indicate a dependency relation between the first and the fourth [target constituent](#), exemplified for the [4NC](#) *church<sub>A</sub> development<sub>B</sub> aid<sub>C</sub> projects<sub>D</sub>* being aligned to the Italian [phrasal equivalent](#) *progetti<sub>D</sub> ecclesiastici<sub>A</sub> di aiuti<sub>C</sub> allo sviluppo<sub>B</sub>* (lit: ‘projects ecclesiastical of aid to development’), as illustrated in Figure 25.7. However, [NSTAP](#) fails to promote the correct [parse tree](#) for [3NCs](#) whose [aligned phrases](#) reveal a dependency relation, such as for the [3NC](#) *world<sub>A</sub> reserve<sub>B</sub> currency<sub>C</sub>* being aligned to the French [phrasal equivalent](#) *monnaie<sub>C</sub> mondiale<sub>A</sub> de réserve<sub>B</sub>*. Both full [parse trees](#) are semantically invalid (cf. Section 25.3.1) and their [subtrees](#) spanning *world reserve* and *reserve currency* have the same *treeScore* factor (cf. Formula 25.7), as shown in Figure 26.1.

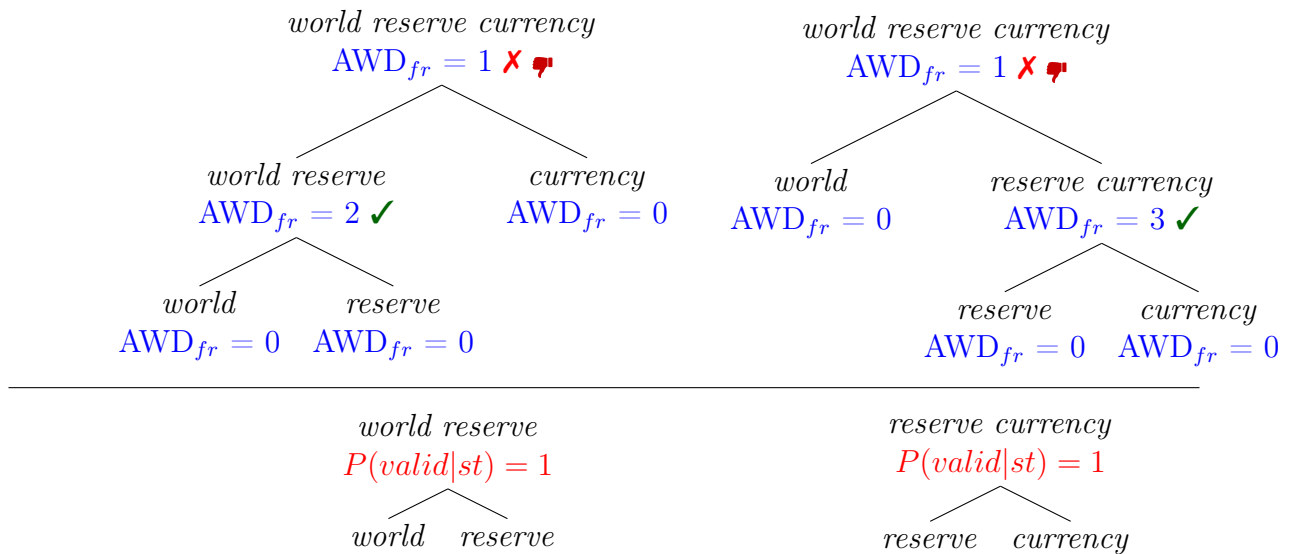


Figure 26.1.: Trees for *world reserve currency*

The reason for this problem is that the [parsing](#) methods presented in Chapter 25 are based on the [adjacency model](#) (23.1.1), i.e., we only consider the [AWDs](#) of adjacent [target constituents](#). For analysing a [3NC](#) A B C in the [dependency model](#), Lauer (1994)

compared the [semantic association](#) between A and B to that between A and C. While the [Aligned Phrase Pattern Parsing \(APPP\)](#) in our pilot study in Chapter 24 can be considered as a hybrid of [AdjMod](#) and [DepMod](#), the development of a [Adjacency-Dependency Model \(AdjDepMod\)](#) for [AWD-based parsing](#) will be addressed in future work. Below, we sketch how such an [AdjDepMod](#) can look like.

In the [AdjDepMod](#), a non-terminal [parse tree](#) node  $N$  is annotated with an [AWD](#) using a recursive function which is applied top-down, as described in Algorithm 26.2.

---

**Algorithm 26.2** [AdjDepMod](#) Annotation Function

---

```

function AdjDepMod\_AWD( $N$ .LEFT,  $N$ .RIGHT)
1: if leaf node( $N$ .RIGHT) then
2:   return AWD( $N$ .LEFT,  $N$ .RIGHT)
3: else
4:    $AWD_X \leftarrow$  AWD( $N$ .LEFT,  $N$ .RIGHT)
5:    $AWD_Y \leftarrow$  AdjDepMod\_AWD( $N$ .LEFT,  $N$ .RIGHT.RIGHT)
6:   if  $AWD_Y \leq AWD_X$  then
7:      $N$ .DEP  $\leftarrow$  TRUE
8:     return  $AWD_Y$ 
9:   else
10:    return  $AWD_X$ 
11:  end if
12: end if

```

---

This function recursively searches for the minimum [AWD](#) between a node's left daughter's [constituent](#) and any right descendant of  $N$ . If the right daughter node of  $N$  is a [leaf node](#),  $N$  is annotated with the [AWD](#) of  $N$ 's [immediate constituents](#),  $N$ .LEFT and  $N$ .RIGHT (lines 1-2). Otherwise, the minimum distance between  $N$ .LEFT and any right descendant of  $N$  is used. Therefore, we recursively compare the [AWD](#) to the direct right daughter's [constituent](#) ( $AWD_X$ ) with the function's value applied to the direct left daughter's [constituent](#) and the right daughter's daughter's [constituent](#) ( $AWD_Y$ ) (lines 4-5). If the latter is smaller (or equal to) the first,  $N$  is annotated with a dependency marker DEP and  $AWD_Y$  is returned as distance between  $N$ .LEFT and  $N$ .RIGHT (lines 6-8), otherwise  $AWD_X$  is returned (line 10).

As example for illustrating the process of the recursive annotation function, we use the [5NC twin pipe undersea gas pipeline](#), whose [parse tree](#) is given in Figure 25.2, repeated in Figure 26.2.

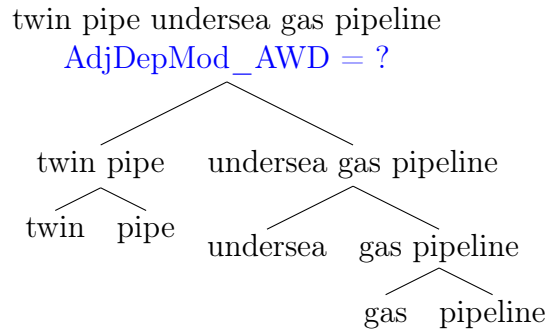


Figure 26.2.: Tree structure for *twin pipe undersea gas pipeline*

For annotating the **root node** in the **AdjDepMod**, it is necessary to compare the **AWD** values between *twin pipe* and *undersea gas pipeline*, between *twin pipe* and *gas pipeline*, and between *twin pipe* and *pipeline*. The final **root node** annotation is the minimum value. If this minimum value does not equal  $AWD(\textit{twin pipe}, \textit{undersea gas pipeline})$ , the **root node** is additionally annotated with **DEP**. The reason for marking the nodes with **DEP** is important for the **parse tree** semantic validation: while the nodes need to have monotonically increasing **AWD** annotations when traversing the tree bottom-up (cf. Section 25.3.1) in the **AdjMod**, in the **AdjDepMod**, the comparison between a node  $N$  and its mother node ( $\text{mother}(N)$ ) is ignored if  $\text{mother}(N)$  is marked with **DEP**.

Finally, we can illustrate that using the **AdjDepMod**, **NFTAP** can provide a semantically valid **RIGHT-branched parse tree** for our initial example *world reserve currency* being aligned to the French **phrasal equivalent** *monnaie mondiale de r serve*, as shown in Figure 26.3.

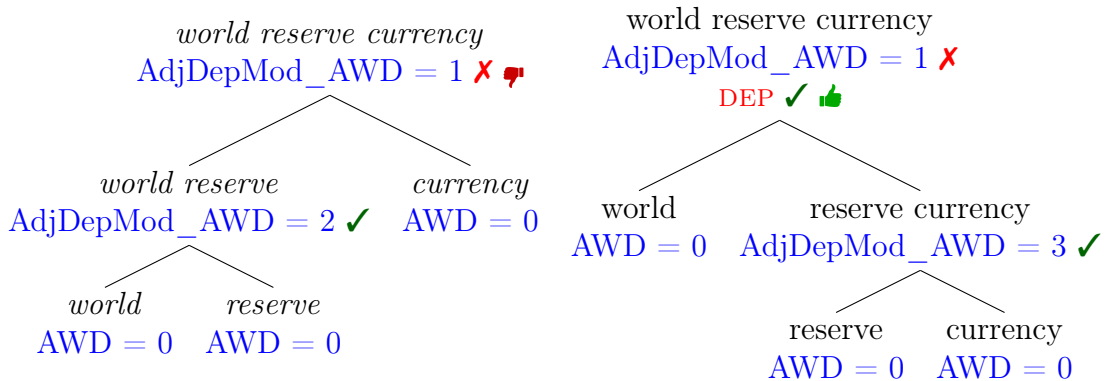


Figure 26.3.: AdjDepMod-annotated parse trees for *world reserve currency*

To the best of our knowledge, this would be the first time, the **dependency model** is directly applied to a **compound size**  $k > 3$ .

### 26.3.5. Revised Dependency Model

In our experiments on [cross-lingual compound parsing](#) we observed that the [adjacency model](#) outperforms the [dependency model](#). As discussed in Section 23.1.2, the main reason for this is that the [DepMod](#) assumes a RIGHT-branched [compound](#) [A [B C]], where both A and B independently modify C. This does not hold for cases in which B C constitute a non-compositional (or at least lexicalized) [compound](#). The fact that there is a [semantic association](#) between A and B C does not necessarily mean that there is a [semantic association](#) between A and C. Previous statistical approaches on [compound parsing](#) based on the [DepMod](#) count bigrams of A C (as adjacent [words](#)).

Considering the issue of the [dependency model](#) above, we propose to use the frequency of dependency relations between A and C in a dependency-parsed training set (for any [3NCs](#) including A and C as [constituents](#)) or the frequency of instances of the pattern A  $c_i$  C (for any valid [constituent](#)  $c_i$ , e.g., a noun).

### 26.3.6. Evaluation Setup for Illustrating the Potential of Token-based Compound Parsing

As discussed in our experiments in Section 25.2.3, [cross-lingual compound parsing](#) works more precisely in a [type-based](#) mode rather than in a [token-based](#) mode. This is counter-intuitive, because a [token-based](#) approach considers context-dependent structural ambiguity. One reason for the moderate lower precision of [token-based compound parsing](#) is the fact that we use the [ENCD](#), based on the [parallel](#) EUROPARL corpus. This domain, the proceedings of the European parliament, does not include much lexical ambiguity.

We propose to use another [parallel corpus](#) for evaluating [token-based](#) vs. [type-based compound parsing](#) and thus prove the potential of our [cross-lingual](#) methods in [parsing](#) structurally ambiguous [compounds](#). As an alternative, the test set can be designed as a balanced selection of structurally ambiguous and non-ambiguous [compounds](#).

Another reason for the lower [token-based](#) precision is that [word alignment errors](#) lead to false [parsing](#) results. In a [type-based](#) mode, these [word alignment errors](#) can be mitigated. For all test set instances, we propose to inspect the alignments to expressive phrases in the [support languages](#) and improve [word alignment errors](#) prior to the application of [cross-lingual compound parsing](#).

### 26.3.7. Adaptation of Cross-lingual Metric-based Compound Parsing on Non-parallel Data

A limitation of the [cross-lingual compound parsing](#) methods presented in this part is the dependency on [parallel corpora](#), which are sparse and frequently domain-specific. This impedes the creation of an end-to-end [parser](#), since no [parse tree](#) can be provided for [compounds](#) that do not occur in the sparse [parallel data](#) at hand.

A desirable strategy is to overcome the dependency on sparse [parallel corpora](#) but still exploit the benefits of [cross-lingual compound parsing](#). As a possible example, Algorithm 26.3 shows the pseudo-code for a method that [cross-lingually parses 3NCs](#) (i.e., LEFT/RIGHT-classification) using non-parallel data but a bilingual dictionary. The algorithm is applied to a [target compound](#) A B C and a [support language](#)  $l_{sup}$ . As input, a bilingual dictionary  $DICT_{l_{tgt} \rightarrow l_{sup}}$  mapping from a [target language](#)  $l_{tgt}$  to  $l_{sup}$ , and a monolingual corpus in  $l_{sup}$  which is segmented in predefined context units is used. As discussed in Section 26.3.3, finding an optimal context unit is based on a trade-off between coverage and precision. In the first step, a counter for all the **translated word distances** (TWDs) of all three [word](#) pairs and a counter for the common context units are initialized with 0 (lines 1-4). In the next step, the method iterates over all possible dictionary translations of all [target constituents](#) (lines 5-7). All common context units for the three translations are collected from the provided monolingual corpus (line 8). The counter `numCommonUnits` is increased by the number of common units for the three current translations (line 9). For all three [word](#) pairs, the TWD counter is increased by the distance of the respective [target constituent](#) translations in the current context unit (lines 11-13). After having processed all translation combinations, the counted TWDs are divided by the number of common units (lines 18-20). Finally, if the smallest TWD is that between A and B, the method returns LEFT, otherwise RIGHT (lines 21-24).

One of the main problems of this approach is that it accumulates [word](#) sense ambiguities from both the [target language](#) and the [support language](#). For example, the English [word](#) *right* is listed with 24 translations in the `dict.cc` dictionary, most of these denoting different senses. Using a small size of the predefined context unit, it might be possible to mitigate this issue with the condition of having a common unit, which can function as [word](#) sense disambiguation means. Another issue is the high runtime complexity of the proposed algorithm, when inspecting the common context units of all possible translations (i.e., four recursively embedded for-loops). Thus, the set of possible translations needs to be filtered (e.g., using a corpus frequency threshold) prior to

collecting common context units.

---

**Algorithm 26.3** Non-parallel approach to cross-lingually parsing a compound A B C

---

**Input 1:** Bilingual dictionary  $\text{DICT}_{l_{tgt} \rightarrow l_{sup}}$

**Input 2:** Monolingual corpus context units for all [support languages](#)  $l_{sup}$

```

1: TWD_A_B  $\leftarrow$  0
2: TWD_A_C  $\leftarrow$  0
3: TWD_B_C  $\leftarrow$  0
4: numCommonUnits  $\leftarrow$  0
5: for translation  $A_{l_{sup}}$  in  $\text{DICT}(A)$  do
6:   for translation  $B_{l_{sup}}$  in  $\text{DICT}(B)$  do
7:     for translation  $C_{l_{sup}}$  in  $\text{DICT}(C)$  do
8:       commonUnits = {u | u includes  $A_{l_{sup}}$ ,  $B_{l_{sup}}$  and  $C_{l_{sup}}$ }
9:       numCommonUnits  $\leftarrow$  numCommonUnits + |commonUnits|
10:      for  $u \in$  commonUnits do
11:        TWD_A_B  $\leftarrow$  TWD_A_B + |pos(DICT(A)) - pos(DICT(B))|
12:        TWD_A_C  $\leftarrow$  TWD_A_C + |pos(DICT(A)) - pos(DICT(C))|
13:        TWD_B_C  $\leftarrow$  TWD_B_C + |pos(DICT(B)) - pos(DICT(C))|
14:      end for
15:    end for
16:  end for
17: end for
18: TWD_A_B  $\leftarrow$  TWD_A_B / numCommonUnits
19: TWD_A_C  $\leftarrow$  TWD_A_C / numCommonUnits
20: TWD_B_C  $\leftarrow$  TWD_B_C / numCommonUnits
21: if TWD_A_B is smallest then
22:   return LEFT
23: else
24:   return RIGHT
25: end if

```

---

### 26.3.8. Exploiting Cross-lingual Supervision for Monolingual Training

While there is a crucial limitation of [cross-lingually supervised](#) methods, i.e., the dependence on [parallel data](#), a way for using the high precision of [cross-lingual supervision](#)

for [compound analysis](#) methods applied on monolingual data is to use the output of [cross-lingually supervised compound analysis](#) as training data for monolingual methods. We will leave this idea to future work.

For the task of [compound parsing](#), we could use our methods (i.e., [DBUP](#) or [NTAP](#)) for training monolingual, supervised [compound parsers](#) such as the methods of Vadas and Curran (2007b) or Pitler et al. (2010). Since both systems are based on out-of-context features of monolingual [NPs](#) (e.g., as given in the [Penn Treebank \(PTB\)](#), annotated by Vadas and Curran (2007a)), we could directly train the supervised methods with feature-value pairs determined for the cross-lingually parsed [compounds](#), as outputted by our [parser](#). By adding our data to the existing training set, the change in performance would illustrate the potential of our [cross-lingual](#) method.

While Vadas and Curran (2007a) annotated a set of **5582** [compound parses](#), which is claimed to be an order of magnitude larger than previous data sets, applying the [non-deterministic full tree accumulation parsing \(NFTAP\)](#) on the [ENCD \(CCR\(0\)\)](#) results in a set of **3668** [cross-lingually](#) parsed [compound types](#), the same order of magnitude as in Vadas and Curran (2007a), which makes the addition of the [cross-lingually](#) parsed [compounds](#) a promising direction for future work.



Part F.

Bottom Line



# 27. Summary, Conclusion and Future Work of the Thesis

## 27.1. Summary of the Thesis

In this thesis, we addressed the task of [compound analysis](#) including the determination of [compoundhood](#) and the [structural analysis](#) of [compounds](#) (i.e., [compound splitting](#) and [compound parsing](#)), as sketched in [Figure 1.2](#), repeated in [Figure 27.1](#).

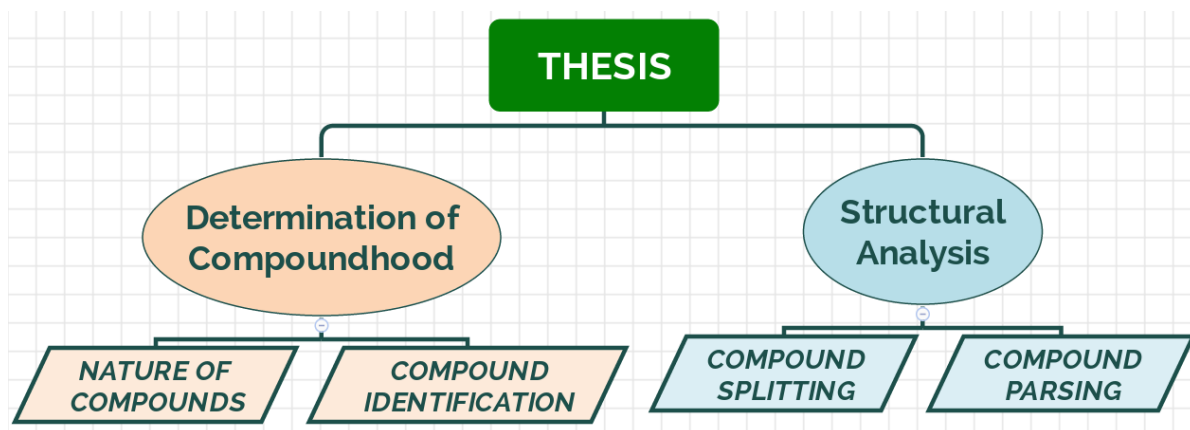


Figure 27.1.: Thematic structure of the thesis

In **Part A**, **Chapter 1**, we introduced the thesis, described the motivation (1.2) of analyzing [compounds](#) (1.2.1) and of our methodology (1.2.2), presented an overview of the main research questions that guided the research process (1.3) and discussed all main contributions that this thesis aims to make (1.4).

In **Part B**, we presented the background for our research subject, i.e., the nature of [compounds](#), as described in linguistics literature. We described all relevant characteristics of [compounds](#) in **Chapter 3**. **Chapter 4** provides a discussion about the controversy of the definition and even the existence of [compounds](#) as given in linguistics literature. An elementary aspect of our way to deal with [compound analysis](#) is the [cross-lingual](#) perspective. This [cross-linguality](#) is motivated by observations about

compounding across languages, discussed in **Chapter 5**.

In **Part C**, we addressed the cross-lingual identification of compounds. At first, we presented previous related work on the identification and discovery of compounds and on different compound resources in **Chapter 8**. The cross-lingual approach presented in this thesis is based on a parallel corpus. As example, we used a part of the parallel EUROPARL corpus. We described this corpus in **Chapter 9**. Before developing the identification method, in **Chapter 10**, we performed two pilot studies, the Linguistic Criterion Inspection (LCI) (10.1), where we collected human ratings for various linguistic criteria for compoundhood and compared their correlation to compoundhood ratings, and the Cross-lingual Compound Inspection (XCI) (10.2), where we explored the most frequent spelling formations of cross-lingual equivalents. **Chapter 11** presented the main method for compound identification. The result of applying the identifier to EUROPARL is the Europarl Nominal Compound Database (ENCD). We outlined the ENCD in **Chapter 12**. Finally, we evaluated the performance of our proposed compound identification method in **Chapter 13**.

In **Part D**, we addressed the first task of the structural analysis of compounds, i.e., compound splitting. Firstly, we outlined previous related work on splitting compounds in **Chapter 16**. An elementary concept in multilingual compound splitting is the Morphological Operation Pattern (MOP). We described the compilation and usage of MOPs in **Chapter 17**. The main method of this part is the recursive binary splitter based on word inflection as an approximation of constituent inflection. We explained the architecture and functionality of this method in **Chapter 18**. Another contribution provided with this part is the re-ranking method with which a frequency-based compound splitter is enriched with information from Distributional Similarity (Dsim). Finally, in **Chapter 20**, we proposed a novel extrinsic evaluation method based on the semantic task of Recognizing Textual Entailment (RTE).

In **Part E**, we addressed the second task of the structural analysis of compounds, i.e., compound parsing. We outlined previous and related work on compound parsing in **Chapter 23**. An elementary aspect of our parsing approach is the cross-lingual perspective. We presented a first pilot study of parsing three-Noun Compounds (3NCs) cross-lingually based on a predefined set of Aligned Phrase Patterns (APPs) (i.e., the Aligned Phrase Pattern Parsing (APPP)) in **Chapter 24**. To avoid the limitation of being restricted to APPs, we defined the aligned word distance (AWD) metric and developed various parsing methods in **Chapter 25** including the deterministic bottom-up parsing (DBUP) (25.2), the non-deterministic full tree accumulation parsing (NFTAP)

(25.3.2) and the [Non-deterministic Subtree Accumulation Parsing \(NSTAP\)](#) (25.3.3).

## 27.2. Conclusion of the Thesis

To conclude this thesis, we review the two main research questions that guided our work. All subquestions of the main research questions are answered in the individual parts of the thesis.

**RQ\_1:** What are [compounds](#)?

⇒ As described in Section 6.2, we follow a perspective of Lieber and Štekauer (2009) saying that there are no clear classes ‘[compound](#)’ and ‘[non-compound](#)’, but instances of more or less compoundlike expressions. In Part C, Chapter 10, we performed two pilot studies in order to reveal the true nature of [compoundhood](#). In the [LCI](#), we observed that there are three [linguistic criteria](#) that correlate best with [compoundhood](#): (1) the [inseparability](#), (2) the [inability to modify the modifier](#) and (3) the [prosody](#). In the [XCI](#), we observed that [compound equivalents](#) can be used for pointing to more compoundlike [targets](#). The more [closed compounds](#) among the [cross-lingual equivalents](#), the higher the degree of [compoundhood](#).

In conclusion, we propose the following indicators for characterizing English [compounds](#). While these indicators are neither necessary nor sufficient, they should be considered as in a graded scale, i.e., the [compoundhood](#) level tends to rise as more indicators are satisfied.

An English [word](#) sequence  $\Psi$  tends to be a [compound](#) if

- (1) no element (e.g., an adjective) can be inserted in  $\Psi$  with preserving the meaning (→ [Inseparability](#))
- (2) the non-final [constituents](#) of  $\Psi$  cannot be modified by external [words](#) (→ [Inability to modify the modifier](#))
- (3) one of the non-final [constituents](#) of  $\Psi$  gets a prosodic stress (→ [Prosody](#))
- (4) there are some [compound equivalents](#) of  $\Psi$ , as given in a [parallel corpus](#) (→ [Cross-lingual spelling](#))

The [target language](#) of our studies ([LCI](#) and [XCI](#)) was *English*. We expect other [criteria](#) to be more relevant for other languages (e.g., the (monolingual) [spelling](#) for *German*).

**RQ\_2:** Does the automatic [analysis of compounds](#) based on [indirect supervision](#) lead to good results?

⇒ We developed three indirectly supervised methods for the automatic [compound analysis](#): a [cross-lingual compound identification](#) method (Part C, Chapter 11), a [multilingual compound splitter](#) (Part D, Chapter 18) and various [cross-lingual compound parsers](#) (Part E, Chapter 24 and Chapter 25).

- ① The [compound identification](#) method is based on [cross-lingual supervision](#), more specifically on the [Closed Compound Restrictor \(CCR\)](#) condition. In the experiments described in Chapter 13, we observed that while the precision steadily increases for higher values of the  $\Xi_{closed}$  parameter, there is a strong recall drop, leading to an overall drop in  $F_1$ -Score. Thus, we can conclude that the [CCR](#) condition is precise but needs to be revised in order to get a higher recall. Despite the recall drop, we still believe that [CCR](#) is a good restrictor, since [compound identification](#) is a hard task for humans too, which is reflected in a moderate [IAA](#), as discussed in Section 10.1.2. The best restrictor, [CCR\(1\)](#), achieved a Jaccard coefficient of 0.236, which is 0,195 below the upper bound of human annotations, and an  $F_1$ -Score of 0.382, which is 0.221 below the upper bound.
- ② The [compound splitting](#) method relies on [supervision based on morphological regularities](#), more specifically on the usage of morphological operations learned from [word inflection](#) as an approximation of [constituent inflection](#). The experiments described in Section 18.6 showed that our approach achieves competitive performance in the discipline of determining the correct [split points](#). For the discipline of [constituent normalization](#), even when compared to language-dependent and knowledge-rich methods, our approach performs well. Its limitations can be mainly attributed to the fact that while there are many operations shared by both [word inflection](#) and [constituent inflection](#), e.g., the addition of plural morphemes and [linking elements](#) (Neef, 2009), the approach of using [any word inflection](#) operation introduces noise (e.g., unrelated [constituent lemmas](#)).
- ③ The [compound parsing](#) method is based on [cross-lingual supervision](#), more specifically on differences in [word distance](#) of [constituent equivalents](#) of a [target compound](#). This approach is based on the assumption that the First Law of Behaghel (1909) is [cross-lingually](#) valid and the [semantic associa-](#)

tion of constituent equivalents can be mapped on the semantic association of the target constituents. The experiments on compound parsing showed that this assumption is true for most cases, leading to a high accuracy in compound parsing. For example, the type-based deterministic bottom-up parsing (DBUP) outperforms the statistical approach based on Chi Squared ( $\chi^2$ ) by 6.2% in accuracy, as discussed in Section 25.2.3. However, the cross-lingual support in parsing relies on expressive cross-lingual equivalents, e.g., phrasal equivalents. If there are no such expressive equivalents available in a given parallel corpus (e.g., there are only compound equivalents or the word distances of constituent equivalents are the same), the target compound cannot be parsed, leading to a lower coverage. A possible solution can be back-off models to statistical or baseline approaches, e.g.,  $DBUP_{type} \rightarrow \chi^2$ , as suggested in Section 25.2.3.

We can conclude that indirect supervision in terms of cross-lingual differences in surface patterns is very helpful in determining the internal structure of compounds (i.e., compound parsing). It is also helpful but too restrictive for compound identification. Indirect supervision stemming from monolingual morphological patterns leads to competitive performance too.

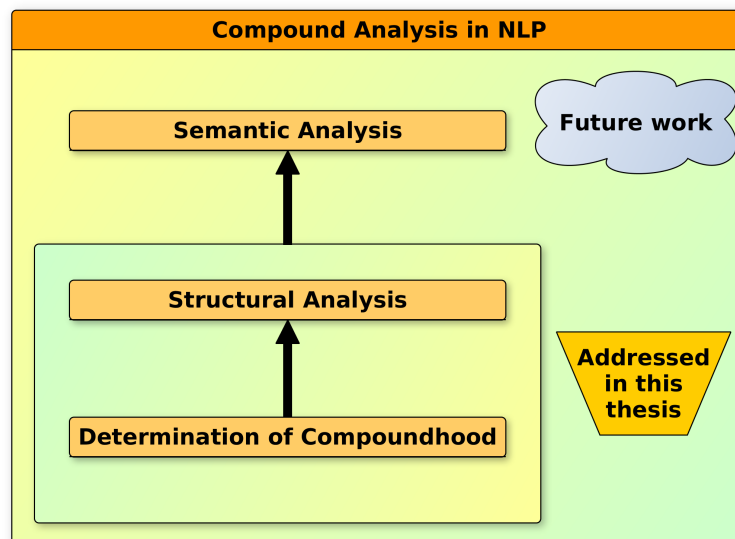


Figure 27.2.: Compound Analysis in NLP

### 27.3. Future Work of the Thesis

In this section, we describe possible overall future work of the thesis. Part-specific ideas for future work were discussed in the respective conclusion chapters.

**Head category:** In this thesis, we focused on the major category of **compounds**, viz. **nominal compounds**. In future work, we will inspect **linguistic criteria** for other PoS (e.g., **adjectival compounds**) and investigate the applicability of all **compound analysis methods** (i.e., **identification**, **splitting** and **parsing**) on minor head categories. We expect to see a similar performance for **adjectival compound** or **verbal compounds**.

**Semantic Analysis:** As described in Figure 1.1, repeated in Figure 27.2, in this thesis, we focused on the determination of **compoundhood** and the **structural analysis** of **compounds**.

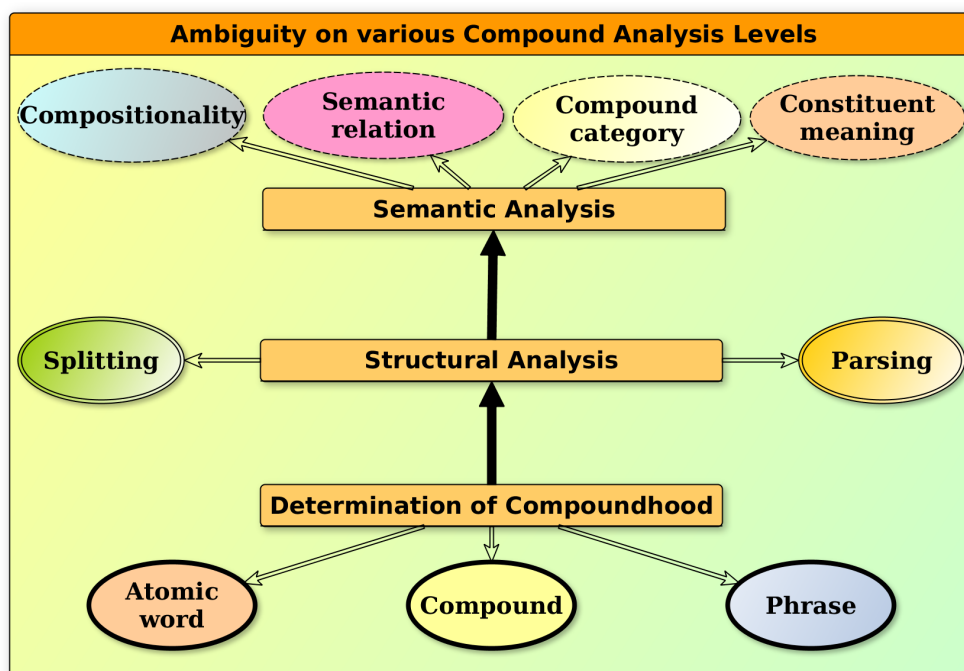


Figure 27.3.: Ambiguity in various Compound Analysis Levels

However, there is still a wide range of ambiguity on the abstract level of semantic analysis (e.g., determining the degree of compositionality, the underlying **semantic relation** or the meaning of all **constituents**), as illustrated in Figure 1.5, repeated in Figure 27.3.



As shown in previous work, semantic ambiguity can be solved using [cross-lingual supervision](#). For example, the determination of the degree of compositionality (Salehi and Cook, 2013), of the underlying [semantic relation](#) (Girju, 2007) or of the [constituent](#) meaning (e.g., using [multilingual WSD](#), Navigli et al. (2013)). We will investigate the potential of [cross-lingual supervision](#) based on the same resource used for [structural analysis](#), i.e., the EUROPARL corpus, on the semantic analysis of [compounds](#) in future work.

**Target languages:** For most parts in this thesis (i.e., the determination of [compoundhood](#) and [compound parsing](#)), we restrict our experiments on English as [target language](#). For the task of [compound splitting](#), (mainly due to the lack of gold standards) we considered only German, Dutch and Afrikaans as [target language](#). However, avoiding manual resources, our approaches are designed language-independently. In future work, we will apply our methods to alternative [open compounding](#) ([identification](#) and [parsing](#)) and [closed compounding](#) ([splitting](#)) [target languages](#).

*27. Summary, Conclusion and Future Work of the Thesis*

Part G.  
Appendix



# A. Universal Surface Patterns

## A.1. Motivation

### A.1.1. Language Independence

Languages often have different **PoS** tag sets. For example, English **PoS** taggers commonly use the Penn Treebank tagset (Marcus et al., 1993), whereas German mostly use the Stuttgart-Tübingen tagset<sup>1</sup> (STTS).

In **USPs** we aim to use language-independent tags for the relevant **PoS** tags, which is appealing when creating an overview of (generalized) **PoS patterns** across languages.

### A.1.2. Complexity of Nouns

In common **PoS patterns**, nominal **closed compounds** are usually represented with the category of the **head**, i.e., as single noun. In **USPs**, we distinguish single nouns (**SN**) from **noun compounds** (**NC**). This has the benefit of exploiting **cross-lingual** evidence for determining the structure of a **kNC**. For example, a **phrasal equivalent** of a **3NC** represented with the **USP SN FC NC** (i.e., a single noun followed by a sequence of **function words** and a **noun compound**) points to a LEFT-branched **3NC**. We exploited this type of information in the **Aligned Phrase Pattern Parsing (APPP)**, presented in Chapter 24.

### A.1.3. Functional Context

While in some tasks of **compound analysis**, the **function words** connecting a **compound's constituent equivalents** in an aligned **support language** are meaningful (e.g., the Romance prepositions for determining the **semantic relation** (Girju, 2007)), for many other tasks of **compound analysis**, it does not matter which preposition or determiner is used in a **phrasal equivalent**. Therefore, in **USPs**, we generalize any sequence of **function words** to the category **functional context** (**FC**). This simplifies the application of **PoS patterns** having various sequences of **function words** in **cross-lingual compound parsing**.

---

<sup>1</sup>[ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html](https://ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html)

## A.2. Transformation of PoS Patterns to USPs

The **USP** is a generalization of a **PoS pattern** of a **cross-lingual equivalent**. It includes information about the complexity of a content word (A.1.2) and conflates a sequence of **function words** into a functional context (FC) (A.1.3).

For a given **word** sequence  $\omega$  including information about **PoS** tags and **split points**, the **generalization function**  $\phi$  produces a **USP**  $\Lambda$  where **content words** are represented by a universal **PoS** tag initialized by the number of **constituents**. All remaining word categories (e.g., determiners, prepositions, pronouns, etc.) are considered as **function words**. The universal **PoS** tag for **content words** only encodes the word class and no further information (e.g., number, gender or case). Some German and Dutch examples of **word** sequences  $\omega$  (first column) and related language-specific **PoS** tags (second column) mapped to a **USP**  $\Lambda$  (third column) are shown in Table A.1.

Word seq $\omega$	PoS pattern	USP $\Lambda$
<i>Mann</i> (man)	NN	SN
<i>dood straf</i> (death penalty)	NN	2NC
<i>van de</i> (of the)	prep det _art	FC
<i>einkommensteuerrechtlichen</i> ({income tax} <sub>ADJ</sub> )	ADJ	3ADJC
<i>rad fahren</i> (to cycle)	VVINF	2VC
<i>Kernarbeitsnormen</i> (core labour standards)	NN	3NC
<i>Rechts vorschriften</i> <i>für die Wettbewerbs politik</i> (legislations for competition policy )	NN APPR ART NN	2NC FC 2NC

Table A.1.: Mapping from words to Universal Surface Patterns

### A.2.1. Simplified USPs

In the scope of this thesis, we focus on **nominal compounds**. Moreover, in **compound identification** and **parsing** of **3NCs**, there is no need for knowing the exact number of composed **constituents** but only whether a noun is simplex (**SN**) or complex (**NC**). Therefore, for the sake of simplicity, we decided to use a simplified version of **USPs**(e.g., 2NC is reduced to NC and 3ADJC is reduced to ADJC).

## B. German Constituent Inflection

Table B.1 shows the 20 most frequent **constituent inflection** operations observed in a **German** corpus study by Langer (1998).

Operation	Description	Example	Freq
$\emptyset$	null operation	<i>Kohl suppe</i> ‘cabbage soup’	22,759
$\oplus s$	<i>s</i> -suffix	<i>Staats feind</i> ‘public enemy’	9637
$\oplus n$	<i>n</i> -suffix	<i>Soziologen kongress</i> ‘sociologist congress’	5307
$\oplus en$	<i>en</i> -suffix	<i>Strau ßen ei</i> ‘ostrich egg’	4316
$\oplus nen$	<i>nen</i> -suffix	<i>Wöchnerinnen heim</i> ‘maternity home’	2610
$\ominus us, \oplus en$	<i>us</i> -trunc., <i>en</i> -suffix	<i>Aphorismen schatz</i> ‘aphorism lexicon’	618
$\ominus um, \oplus en$	<i>um</i> -trunc., <i>en</i> -suffix	<i>Museen verwaltung</i> ‘museum administration’	348
$\ominus um, \oplus a$	<i>um</i> -trunc., <i>a</i> -suffix	<i>Aphrodisiaka verkäufer</i> ‘aphrodisiac seller’	255
$\ominus e$	<i>e</i> -trunc.	<i>Kirch hof</i> ‘churchyard’	122
$\ominus a, \oplus en$	<i>a</i> -trunc., <i>en</i> -suffix	<i>Madonnen kult</i> ‘Madonna worship’	95
$\oplus e$	<i>e</i> -suffix	<i>Hunde halter</i> ‘dog owner’	87
”, $\oplus e$	Umlaut, <i>e</i> -suffix	<i>Gänse klein</i> ‘goose giblets’	73
$\ominus on, \oplus en$	<i>on</i> -trunc., <i>en</i> -suffix	<i>Stadien verbot</i> ‘stadium ban’	59
$\oplus es$	<i>es</i> -suffix	<i>Geistes haltung</i> ‘attitude’	43
”, $\oplus er$	Umlaut, <i>er</i> -suffix	<i>Blätter wald</i> ‘leaf forest’	38
$\ominus en$	<i>en</i> -trunc.	<i>Süd wind</i> ‘south wind’	33
$\ominus on, \oplus a$	<i>on</i> -trunc., <i>a</i> -suffix	<i>Pharmaka analyse</i> ‘pharmaceutical analysis’	28
$\oplus er$	<i>er</i> -suffix	<i>Geister stunde</i> ‘witching hour’	25
$\oplus ien$	<i>ien</i> -suffix	<i>Prinzipien reiter</i> ‘stickler for principles’	19
$\ominus e, \oplus i$	<i>e</i> -trunc., <i>i</i> -suffix	<i>Carabinier schule</i> ‘carabiniere school’	11

Table B.1.: German constituent inflection operations (Langer, 1998)

*B. German Constituent Inflection*



# C. Split Point Format Compilation

In this chapter, we describe some ways of compiling the [split point format \(SPF\)](#) from resources where there is no information about [constituent forms](#) but only about [constituent lemmas](#). For evaluating the determination of [split points](#) (as discussed in Part D), there is need for compiling an [SPF](#).

## C.1. Linear Compilation of the Split Point Format

---

**Algorithm C.1** Linear [SPF](#) compilation

---

**Input 1:** [target compound](#)  $c$

**Input 2:** [lemmas](#) {list of [constituent lemmas](#), resulted from splitting  $c$ }

```
1: forms  $\leftarrow$  [ ] {the final constituent forms for the SPF}
2: stem  $\leftarrow c$  {the stem is initialized with the full compound  $c$ }
3: while |lemmas| > 1 do
4:   lastLemma = lastElement(lemmas)
5:   if stem.endsWith(lastLemma) then
6:     forms  $\leftarrow$  lastLemma + forms {last lemma is prepended to forms}
7:     stem  $\leftarrow$  stem - lastLemma {last lemma is truncated from the stem}
8:   else
9:     for suffix of stem do
10:      score(suffix) = len(suffix) / ED(lastLemma, suffix)
11:    end for
12:    inflForm = suffix(maxScore) {get suffix with maximum score}
13:    forms  $\leftarrow$  inflForm + forms {inflected form is prepended to forms}
14:    stem  $\leftarrow$  stem - inflForm {inflected form is truncated from the stem}
15:  end if
16:  lemmas  $\leftarrow$  lemmas - lastLemma {last lemma is removed from the lemmas}
17: end while
18: forms  $\leftarrow$  stem + forms {stem as first constituent form}
19: return join(|,forms) {creation of the final SPF}
```

---

Some of the [compound splitting](#) gold standards and [compound splitters'](#) output provides only an [LSF](#) but no [SPF](#). For compiling an [SPF](#) from a given [target compound](#)

and an LSF, we developed a method that iteratively truncates the potential **constituent forms** from the end of the compound.

This method is based on the assumption that **constituent inflection** only comprise **word-internal** (e.g., Umlautung) and **word-final** operations (i.e., suffixation) but no **word-initial** operations (i.e., prefixation). This assumption holds for all languages inspected in this thesis (see Section 3.9), but needs to be adapted for languages which realize **constituent inflection** using prefixation.

Algorithm C.1 shows the pseudo code for the linear SPF compilation. The input is the **target compound**  $c$  and the list of **constituent lemmas**, `lemmas`, provided by the **compound splitter** or by the gold standard. A list of **constituent forms**, `forms` to be determine is initialized (line 1) and a **compound stem**, `stem`, subject to form truncation, is initialized with  $c$  (line 2). While there are more than one **lemma** left, the method checks whether the last **lemma**, `lastLemma`, is a suffix of  $c$ . If so, `lastLemma` is truncated from `stem` and prepended to `forms` (lines 5-7). If not, the method has to deal with a case of **word inflection** or **constituent inflection**. Each suffix of `stem`, `suffix`, is scored with its length divided by the **ED** between `lastLemma` and `suffix` (lines 9-11). The highest-scored `suffix` is used as **constituent form** for `lastLemma` and prepended to `forms` as well as truncated from `stem` (lines 12-14). After processing `lastLemma`, it is removed from the list of **constituent lemmas** (line 16). After processing all but one **lemma**, the resulting **compound stem** is prepended to `forms` (line 18). The final **SPF** is the concatenation of all collected **constituent forms**, separated by | (line 19).

As will be shown in Section C.3, this procedure has a very high accuracy and the few **SPFs** compiled in a wrong way have no relevant impact on the performance numbers presented in the experiments of Section 18.6.

## C.2. Hierarchical Compilation of the SPF

As described in Chapter 18, the **MOP-based compound splitting** method performs a **recursive lemma splitting**, i.e., the binary **splitter** is recursively applied to the normalized **constituents** (instead of on the **constituent forms**). However, this procedure poses a challenge for the compilation of the **SPF** for compounds having three or more **constituents**, as illustrated in Figure C.1, where  $cform$  is the **constituent form** and  $clem$  is the **constituent lemma**, both **relative** to the **constituent lemma** of the mother’s tree node.

### C. Split Point Format Compilation

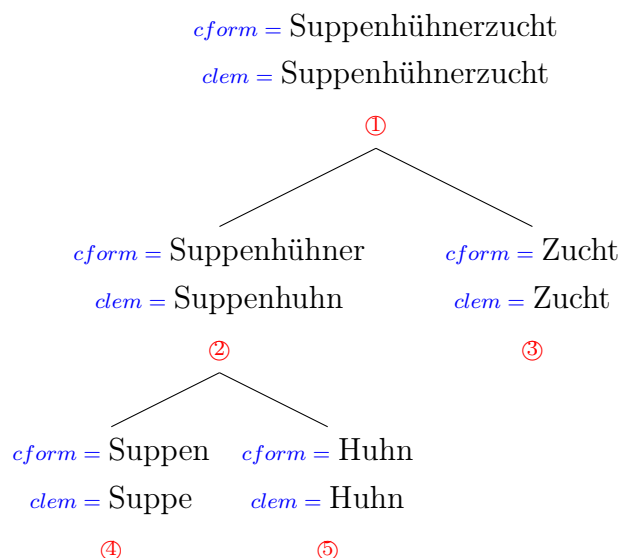


Figure C.1.: Example of a **split tree** with recursive **lemma** splitting

The main issue is based on the fact that  $cform = clem$  for the node ⑤. So, how can we propagate the **split point** in *Suppen | huhn* to the **constituent form** *Suppenhühner* (for arriving at the **SPF** *Suppen | hühner*)? Therefore, we propose two further methods, which will be compared in Section C.3.

#### C.2.1. SPF by using MOP Application

For compiling the **SPF**, the recursive function (as defined in Formula C.1) is applied to the root node of the **target compound's split tree**.

$$SPF(N) = \begin{cases} cform(N) & \text{leaf}(N) \\
 SPF(N.LD) | SPF(N.RD)[head/MOP_{clem(N) \rightarrow cform(N)}(head)] & \text{else} \end{cases} \quad (C.1)$$

If the node  $N$  is a **leaf node**, the **SPF** is the **constituent form** of  $N$ . Otherwise, the **SPF** is the concatenation of the left daughter's ( $N.LD$ ) **SPF**, a pipe symbol and the right daughter's ( $N.RD$ ) **SPF**, in which the **head** (i.e., the rightmost element, separated by a pipe symbol) gets inflected by application with the **MOP** used for transforming the  $clem$  of  $N$  to the  $cform$  of  $N$ .

For example, in Figure C.1, node ②, the **MOP** for transforming *Suppenhuhn* to *Suppenhühner*, is  $u/\ddot{u}:\$/er\$/$ . The **head** of the **SPF** in node ⑤ is the **constituent form**, *huhn*. As a result, the **SPF** of node ② is ' $SPF(④) | SPF(⑤)[huhn/hühner]$ ' = *Suppen | hühner*.

### C.2.2. SPF by using Linear Approach for Constituent Forms

The second hierarchical method is to apply the linear SPF compiler (presented in Section C.1) to the complete compound, but use the recursively collected *cform*'s instead of the constituent lemmas. For example, for the split tree shown in Figure C.1, the linear SPF compiler is applied to the root node's constituent form, *Suppenhühnerzucht*, and the list of mother-node related *cforms*: [*Suppen*, *Huhn*, *Zucht*].

## C.3. Experiments

All necessary information about the experiments' setup is given in Section 18.6.

### C.3.1. Linear SPF Compilation

As described in Section 18.6, some gold standards and the output of some splitting systems lack SPFs, but only provides information for LSFs. Algorithm C.1 shows a way how to compile an SPF given the target compound and a list of constituent lemmas. While the truncation of head lemmas (if matching with the end of the target compound) has perfect accuracy, the constituent form determination using minimum ED between the head lemma and any target compound suffix is slightly less accurate.

In this experiment, we measure the accuracy of the linear SPF compilation (1) for the SPFs in the system output of FF2010 and (2) for the SPFs of compounds without Umlauts in the gold standard of Marek (2006), M2006GS.

**Split Point Format for FF2010** When transforming the splitting output of FF2010 into the evaluation format, we compare the best split of each target compound, annotated with an SPF (as provided by Fritzinger and Fraser (2010)), with our linearly compiled SPF. Table C.1 shows the accuracy for each of the three German gold standards.

Gold standard	Size	# SPF mismatches	Accuracy
HH2011GS	54,148	8	99.9852%
M2006GS	139,081	15	99.9892%
HB2008GS	687	0	100%

Table C.1.: Evaluation of the linear split point format compilation for FF2010

This indicates that the quality of the linear SPF compilation is almost perfect. Moreover, inspecting the SPF mismatches, it turns out that some of the SPFs provided

### C. Split Point Format Compilation

by FF2010 are not correct. For example, while the linear **SPF** compiler produces *wett|tauchen* for the compound *Wetttauchen* ‘diving race’ and the **constituent lemmas** *wetten* ‘to bet’ and *tauchen* ‘to dive’, FF2010 provides the incorrect **SPF** *wetten|tauchen*. Actually, when discounting wrong **SPF** mismatches, the linear **SPF** compilation method even achieves accuracies of **99.9945%** (HH2011GS) and **99.995%** (M2006GS).

**Split Point Format for Compounds of M2006GS** When transforming the morphological parse of M2006GS into the evaluation format, it is possible to produce a gold **SPF** for those (inflected) compounds which do not contain an Umlaut, i.e., for those for which **constituent inflection** and **word inflection** is realized only by **word-final** (i.e., subtractive and/or additive) operations. For these compounds, the gold **SPF** is compared to the linearly compiled **SPF**. For a total of **142,759** entries in the evaluation format having a **target compound** without Umlauts, there are **80 SPF** mismatches, leading to an accuracy of **99.944%**. One reason for the slightly worse accuracy (compared to the accuracy for FF2010) is the fact that for the **SPFs** of FF2010, we use **word-inflected head lemmas**, whereas for the **SPFs** of M2006GS, we use **head lemmas** without any inflection. For **word-inflected target compounds**, a non-inflected **head lemma** produces more noise.

**Conclusion** In general, the performance quality of the linear **SPF** compilation is very high. Errors in the **SPF** compilations should have no relevant impact on the differences between the **splitting** systems compared in the experiments in Section 18.6.

#### C.3.2. Hierarchical **SPF** Compilation

In Section C.2, we mentioned two methods for compiling an **SPF** given a **split tree** annotated with **constituent lemmas** and **constituent forms**, that are directly derived from the mother node (as shown in Figure C.1). In this experiment, we compare the **SPX** numbers for the **MOP**-based **compound splitter** being applied to M2006GS with all three different **SPF** compiling methods, as shown in Table C.2.

Altogether, there is hardly any difference in these three **SPF** compilers. The best performance is achieved with the linear approach applied to the **constituent forms**. In an error analysis, some minor individual advantages and disadvantages have come out.

As discussed in Section 17.3, during application of **MOPs**, the position of the **word**-internal operations is underspecified and for the case of several possible replacement positions, the last possible is used by default. While this convention is valid in most cases, a false lemmatization can yield misleading **MOPs** for which **MOP** application

### C. Split Point Format Compilation

SPF compiler	SPX			
	<i>P</i>	<i>R</i>	<i>Acc</i>	<i>F</i> <sub>1</sub>
MOP application (C.2.1)	97.2	96.4	96.8	96.8
Linear on forms (C.2.2)	<b>97.3</b>	96.4	96.8	96.8
Linear on lemmas (C.1)	97.2	96.4	96.8	96.8

Table C.2.: Evaluation of the different split point format compilers

results in the false **constituent form**. For example, the pluralized compound *Frei|gaben* ‘releases’ (lit. ‘free givings’) is not split into the **constituent lemmas** *Frei* and *Gabe*, because the latter is usually found as compound **head** only. Instead, the **head** is plausibly normalized to the verb *geben* ‘to give’ (i.e., *gaben* is a valid plural past tense form of *geben*). The resulting **MOP** for transforming *freigeben* to *freigaben*, **e/a**, can be applied with two possible positions in the **head**, *geben*. Using the aforementioned convention, it is applied to the last possible position, leading to the false **head** form *geban*.

On the other hand, the usage of **MOP** application can overcome some limitations of determining the suffix with the smallest **ED** (C.1). For example, for the pluralized **compound** *Kuckuckseier* ‘cuckoo’s eggs’, the **MOP** from the **compound lemma**, *Kuckucksei*, is **\$/er\$**. Applying this **MOP** to the **head**, *ei*, yields the correct **head** form *eier*. In contrast, the suffix of *Kuckuckseier* with the minimum **ED** to *ei* is *er* (**ED** = 1) and not *eier* (**ED** = 2).

While the assumption that **constituent inflection** is commonly realized with **word-internal** and **word-final** operations, is valid, it is not always correct for **word inflection**, occurring on the **compound head**. Thus, when using the **head’s lemma** for the linear **SPF** compilation, the minimum **ED** may be only a suffix of the correct **head** form. For example, for the complex participle *fein|gebrannt* ‘fine-fired’, the **head lemma** is *brennen* ‘burn’. The determined **head** form, i.e., the suffix with the minimum **ED**, is *brannt* instead of *gebrannt*.

**Conclusion** In general, there are hardly any differences in the three proposed **SPF** compilers. Nevertheless, for the experiments presented in Section 18.6, the linear **SPF** compiler based on the mother-node related **constituent forms** is used, because it provides the highest precision, **SPP**, as shown in Table C.2.

## C.4. SPF Compilations of Resources

### C.4.1. SPF for HH2011GS

Since the **target compounds** are not inflected (e.g., pluralized), the application of the linear **SPF** compilation (outlined in Algorithm C.1) always triggers the **head truncation** (providing 100% accuracy), i.e., the **constituent form** of the **modifier** is the result of truncating the **head lemma** from the target compound. The **split point** is set between **modifier** form and **head lemma** (e.g., *Hühner* | *fleisch*).

### C.4.2. SPF for M2006GS

For compiling the **SPF**, the **linking element** and **word**-inflected suffixes from the morphological parse are added. However, the morphological parses do not indicate any **word**-internal operations (e.g., Umlautung). For example, the **split** of *Hühnerfutter* ‘chicken feed’ is represented as `huhn|er{n}+futter{n,v}`. In this case, the **constituent form** of *Huhn* would be *Huhner* (instead of *Hühner*).

Thus, we selected the linear **SPF** compilation (outlined in Algorithm C.1), if the **target compound** contains an Umlaut; otherwise, the **SPF** derived from the morphological parse is used.

### C.4.3. SPF for FF2010

The system of Fritzinger and Fraser (2010) produces an **SPF** only for the best split but not for all proposed **compound splits** in the ranking. Since this best split (which exclusively provides an **SPF**) does not necessarily have  $k_{gold}$  **constituents**, it is not always possible to get an **SPF** from the system, because we select the highest-ranked **compound split** having  $k_{gold}$  **constituents**. Therefore, we decided to compile the **SPF** for all proposed **LSFs** and for all values of  $k$  in the ranking output using the linear **SPF** compilation presented in Algorithm C.1.

Using the **SPFs** of the best splits in the system output of Fritzinger and Fraser (2010) as gold **SPF**, the linear **SPF** compilation can be evaluated, as shown in Section C.3.

### *C. Split Point Format Compilation*



# D. Further Compound Splitting

## Gold Standards

There are some German gold standards for [compound splitting](#) that were not addressed in the experiments presented in Section [18.6](#).

### D.1. German Splitting Gold Standard of Cap (2014)

Cap (2014) developed a compound splitting gold standard (C2014GS) derived from the development set of the 2007 Workshop on Statistical Machine Translation<sup>1</sup>. It includes both samples of true [compounds](#) and particle verbs as well as [atomic words](#).

The samples are restricted to the [compounding word](#) formation and exclude derivational processes like suffixation or prefixation. In other words, derivational cases like *Untersuchungshäftling* ‘prisoner awaiting trial’ are only included as [atomic words](#), which are kept unsplit, although a systematic [compound splitter](#), which lacks a deep morphological analysis for distinguishing [compounding](#) and derivational processes (as given by morphological analyzers such as SMOR), would split the [word](#) into the [constituents](#) *Untersuchung* ‘investigation’ and *Häftling* ‘prisoner’. Moreover, we consider the [compoundhood status](#) of *Untersuchungshäftling* as debatable. Cap (2014) argues that there would be a semantic shift when [splitting](#) *Untersuchungs|häftling*, i.e. while the derivational meaning denotes a ‘person being in investigative custody’, the [compound](#) meaning denotes a ‘prisoner under investigation’. Besides the fact that Cap (2014) associates the [compounding](#) analysis with a plausible interpretation, while not performing any approach of [Word Sense Disambiguation](#) (WSD), we do not agree with the described semantic shift but would break it down to the [word](#) sense ambiguity of *Haft* ‘custody / imprisonment’, from where *Häftling* is derived of. Finally, in the [Europarl Nominal Compound Database](#) (ENCD), described in Chapter [12](#), the pluralized [compound](#) *Untersuchungshäftlinge* has the English [equivalent](#) ‘prisoners on remand’.

---

<sup>1</sup><http://www.statmt.org/wmt07/shared-task.html>

## D. Further Compound Splitting Gold Standards

Moreover, C2014GS contains particle verbs, which are not subject of the [compound splitter](#) proposed in this thesis. As a consequence, we decided to exclude C2014GS from the experiments described in Section 18.6.

### D.2. Ghost-NN

Schulte im Walde et al. (2016) created a novel German gold standard of noun-noun [compounds](#) called **G<sub>h</sub>ost-NN**, comprising 868 [compounds](#) annotated with different features, such as corpus frequency of the [compound](#) and their [constituents](#), productivity, [word](#) sense ambiguity of the [constituents](#) or compositionality ratings between [compound](#) and both [constituents](#).

	HH2011GS	HH2011GS
	M2006GS	M2006GS
	HB2008GS	HB2008GS
<b>G<sub>h</sub>ost-NN</b>	743	125
<b>G<sub>h</sub>ost-NN</b>	176,619	—

Table D.1.: Splitting gold standard overlap between HH2011GS+M2006GS+HB2008GS and **G<sub>h</sub>ost-NN**

Table D.1 shows the overlap of **G<sub>h</sub>ost-NN** with the union of HH2011GS, M2006GS and HB2008GS. All four gold standards have 743 [compounds](#) in common and there are only 125 samples in **G<sub>h</sub>ost-NN**, which are not covered by the three gold standards used in our experiments. This which means that the majority of **G<sub>h</sub>ost-NN** is already covered by the other three gold standards. Thus, we decided to exclude this gold standard from the experiments presented in Section 18.6.

# E. Annotation Guidelines for Creating the Europarl Nominal Compoundhood Ratings

## E.1. Introduction

For an ongoing experiment, we are collecting **English nominal compounds** as they occur in written text. In addition, we would like to get an idea of their characteristics with respect to several **linguistic criteria** for compoundhood.

It is your task to find **English word sequences** in context that constitute **nominal compounds** (i.e., compounds with a nominal head, which is usually the last element of a compound). Although there is no commonly accepted definition for compounds, an abstract version is given by Bauer (2003):

*“a compound is the formation of a new lexeme by adjoining two or more lexemes”*

We follow the conclusion of Lieber and Štekauer (2009: chap. 1) saying that

*“compounding is a gradient, rather than a categorical phenomenon, with prototypical examples and fuzzy edges”*

In Section [E.2](#), we provide a **list of linguistic criteria** for compoundhood, that could help you to decide whether a word sequence can be considered as a nominal compound.

These tests are extracted from the **first chapter** of the **Oxford Handbook of Compounding**, edited by Lieber and Štekauer (2009), which is attached to these guidelines and provide an insight into the difficulty of defining compoundhood.

**Please read this chapter before starting the annotations!**

In your annotation task, you have to provide ratings for **each linguistic criterion** and for the **compoundhood** of the word sequence under consideration.

## E.2. Rating of Linguistic Criteria for Compoundhood

Although Lieber and Štekauer (2009: chap. 1) come to the conclusion that there is almost no reliable and universally accepted criterion for compoundhood, they discussed several plausible tests that are to be rated in the underlying annotation task. More details about these linguistic criteria, outlined below, can be found in Lieber and Štekauer (2009: chap. 1).

**1. Closed vs. open compounding:** While the components of phrases are usually separated by white space (e.g., *black bird*), compounds can be written as a one-word (closed) compound (e.g., *blackbird* as opposed to the corresponding phrase). However English compounds can have various spelling forms: closed compounds (e.g., *flowerpot*), hyphenated compounds (e.g., *flower-pot*) or also open compounds (e.g., *flower pot*).

⇒ Does the spelling of the expression under consideration (i.e., closed or open compounding) point to compoundhood?

[3] Yes - it is a one-word construction  
(e.g., *blackbird* or *thermo-insulation*)

[2] Partially - it is a multi-word construction but includes a closed compound  
(e.g., *database connection*)

[1] No - all constituents are separated by white space  
(e.g., *energy efficiency action plan*)

**2. Inseparability:** No element should intervene a compound's constituents. While *black bird* can be understood as a compound, *black ugly bird* is a phrase.

⇒ Can you think of a way to insert an element between the constituents of the underlying expression?

[3] No, this is a fixed inseparable expression; inserting any elements would change its meaning (e.g., *French (\*diligent) teacher*)

[2] Not sure, an insertion might be possible without changing the meaning.

[1] Yes, inserting one or more elements is possible and does not change the meaning  
(e.g., *red or black nice angry birds*)

**3. Inability to modify the modifier:** In a phrase like *social person*, the modifier (usually the non-last element) can be further modified (i.e., *very social person*). Commonly, this does not happen for compounds (e.g., (✗ *very*) *social policy*).

⇒ Is there a modifying adjective/adverb or can you think of such an element in the surrounding context that modifies any modifier in the expression under consideration?

[3] No, there is no such modifying element and I can only think of such elements that modify the head or changes the meaning

(e.g., big *computer shop* refers to a big shop, not to a shop selling big computers)

[2] There might be such a modifying element, but it could be a case of lexicalization

(e.g., long *life expectancy*)

[1] Yes, there are plenty of possible elements modifying a modifier

(e.g., (*very|dark|light|...*) *brown dog*)

**4. Inability to replace the head by the pronoun *one*:** In a phrase you can usually replace the head noun by the pronoun *one* (e.g., a *black dog* → a *black one*). This should not happen for compounds (i.e., *blackbird* → *black one* ✗).

⇒ Can you replace the head of the expression under consideration by the pronoun *one*?

[3] No, replacing the head by *one* is absolutely impossible

(e.g., a *biology teacher* → ✗ a *biology one*)

[2] In some marked sentences, such a replacement is possible, but usually not

(e.g., ... a *riding horse*, ... *the carriage ones*)

[1] Yes, the head can be replacement by *one* (e.g., a *brown chair* → ✓ a *brown one*)

**5. Inflection of the modifier:** In a compound, the modifier does not undergo any inflectional operation (e.g., pluralization) but only the head - even if the modifier has a plural interpretation

(e.g., *shoe salesman*). In a phrase, both parts are inflected (e.g., *salesman for shoes*).

⇒ Is any modifier inflected in the expression under consideration?

- [3] No, a (possible) plural marker is only visible with the head  
(e.g., *mouse holes* ✓ vs. *mice holes* ✗)
- [2] It could be possible that the modifier is plural-marked, but this seems like an exceptional case (e.g., *programs coordinator*)
- [1] Yes, there is an inflected modifier (possible) in the expression under consideration  
(e.g., *traps for mice*)

**6. Prosody:** While in a phrase such as *black bird*, the head (i.e., *bird*) is stressed (or both parts have equal stress), in a compound such as *blackbird* the primary stress is commonly on the modifier (i.e., *black*).

⇒ How would you stress the expression under consideration?

- [3] The primary stress is on a modifier (e.g., *French teacher*)
- [2] The primary stress is on the head, but this stress pattern is justified, e.g., by semantics  
(e.g., *iron door*)
- [1] The constituents are equally stressed or primary stress is on the head without obvious justification (e.g., *French teacher*)

### E.3. Please note

Do not forget to also annotate both **open** (e.g., *data base*) and **closed noun compounds** (e.g., *network*).

Sometimes, a noun compound is nested in another noun compound, e.g., *greenhouse gas emissions*. In this case, please annotate both the **complete compound** (i.e., *greenhouse gas emissions*) and **all subordinated compounds** (i.e., *greenhouse gas* and *greenhouse*) separately, as will be shown in Figure E.2.

If a word sequence occurs more than once in a sentence, annotate **all instances** that constitute a compound (and indicate the position as comment).

## E.4. Annotation process

1. You are given a set of **English sentences** stored in a **color-highlighted OpenOffice spreadsheet** as shown in Figure E.1.

ID	SENTENCE	COMPOUND	COMPOUND RATING	1. SPELLING	2. INSEPARABILITY	3. MOD-MOD	4. ONE-REPLACEMENT	5. INFLECTION	6. PROSODY	COMMENTS
ep-98-03-10.xml.gz:3221:0:46	We believe that this programme, as changed by our amendments, will play a fundamental role in protecting the health of the citizens of Europe, by significantly reducing the number of deaths and the financial losses caused by the									
ep-08-02-18-019.xml.gz:195:0:38	I consider it to be the European Union's duty, as a democratic entity, to promote respect for the rights of all the Union's citizens by initiating European programmes of education and information									
ep-11-05-12-013.xml.gz:1783:0:2	At the same time, we should consider introducing cultural visas for artists and for all those working in the cultural sector.									
ep-06-10-23-021.xml.gz:126:0:24	However, the Commission will bear the issues in question in mind within the framework of the development of our policy in this sector.									
ep-01-09-05.xml.gz:1763:0:83	The plain fact is that the majority of this Parliament - the same Parliament that adopted preventive 'apartheid' measures against the new, democratically elected Austrian government, that serious									
ep-08-04-22-016.xml.gz:438:0:5	It is really quite amazing.									
ep-05-09-07.xml.gz:1996:0:17	Indeed, instead of giving it due incentives sometimes we are almost encouraging it to leave Europe.									
ep-00-11-30.xml.gz:78:0:32	Before the Erika sank, many other ships sank as well, with the loss of oil, other cargoes, or tragically, the loss of lives of crew or passengers.									
ep-01-12-11.xml.gz:2135:0:21	Which of the existing mechanisms does the Commission intend to reinforce in order to improve the effectiveness of the Euro-Mediterranean partnership?									
ep-02-02-27.xml.gz:2096:0:15	At present, promoting social and economic development in these states is undoubtedly the imperative.									
ep-11-04-07-003.xml.gz:439:0:7	Some very valid questions have been asked.									
ep-08-07-10-011-02.xml.gz:55:0:	As a member of the South Asian Association for Regional Cooperation ( SAARC ) delegation of the European Parliament, I have visited Bangladesh several times.									
ep-07-07-11-018.xml.gz:41:0:29	Finally, clear political support must be given to every country that uses a flexibility instrument, whatever it may be, which is not the case in practice.									

Figure E.1.: OpenOffice spreadsheet for nominal compound annotation

The columns of the spreadsheet are designed for the following information:

- Column A:** a system-internal ID
- Column B:** the English sentence in which compounds are to be searched
- Column C:** the observed compounds
- Column D:** rating for the observed compound
- Column E:** rating for the linguistic criterion **1. Closed vs. open compounding**
- Column F:** rating for the linguistic criterion **2. Inseparability**
- Column G:** rating for the linguistic criterion **3. Inability to modify the modifier**
- Column H:** rating for the linguistic criterion **4. Inability to replace the head by the pronoun *one***
- Column I:** rating for the linguistic criterion **5. Inflection of the modifier**
- Column J:** rating for the linguistic criterion **6. Prosody**
- Column K:** optional comments for the annotation (e.g., ideas for novel criteria)

2. Please **scan the presented sentence** in column B.
3. If you find a word sequence which constitutes a nominal compound, add it in **column C of the subsequent row**. The rows with sentences remain untouched.

## E. Annotation Guidelines for Creating the Europarl Nominal Compoundhood Ratings

- Assign a **compound rating** (column D) to the selected word sequence in column C:

- [3]** very compoundlike (*i.e., a prototypical compound*)
- [2]** rather compoundlike (*i.e., probably a compound*)
- [1]** mildly compoundlike (*i.e., could be considered as a compound*)

- Afterwards, annotate all subsequent columns (columns E to J) with the **ratings for the linguistic criteria** described in Section E.2.

- Write **one compound per row** and start with a **new row for subordinated compounds**.

If you find more than two compounds in the presented sentence, **add further rows** before the subsequent sentence's row.

Some examples are shown in Figure E.2.

ID	SENTENCE	COMPOUND	COMPOUND RATING	1. SPELLING	2. INSEPARABILITY	3. MOD-MOD	4. ONE-REPLACEMENT	5. INFLECTION	6. PROSODY	COMMENT
293	ep-06-12-14-003.xml.gz:219:0:21	Commission's proposal to encourage public procurement of clean and efficient vehicles, including using high biofuel blends.								
294		public procurement	1	2	2	2	1	3	1	
295		biofuel blends	2	2	3	3	3	3	2	
296		biofuel	3	3	3	3	3	3	3	
297	ep-07-07-10-021.xml.gz:165:0:34	So our offer could be that we will support a CO2-free coal-fired power plant with our technologies, as a gesture, but in return we can demand that our patent rights be respected.								
298		power plant	3	2	3	3	3	3	3	
299		patent rights	3	2	3	3	3	3	3	

Figure E.2.: Examples of spreadsheet-based nominal compound annotation

Finally return the processed document to Patrick.Ziering@ims.uni-stuttgart.de.

## E.5. Training and annotation stage

- We will start with a set of **20 English sentences for training**.
- You are able to ask questions about the annotation task, if you are unsure.
- Return the processed document to Patrick.Ziering@ims.uni-stuttgart.de.
- If everything is clear, you are given more data (probably in batches of 100 sentences) for **annotation**.



# List of Abbreviations

[Symbols](#) | [A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [I](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [W](#) | [X](#)

## Symbols

$\chi^2$

Chi Squared. [286](#), [291](#), [292](#), [298](#), [299](#), [301](#), [322–325](#), [341](#), [348](#), [365](#)

## 2NC

two-Noun Compound. [3](#), [5](#), [8](#), [13](#), [17](#), [30](#), [33](#), [45](#), [80](#), [83](#), [85](#), [87](#), [88](#), [98–101](#), [103–105](#), [131–135](#), [150](#), [152](#), [154](#), [237](#), [265](#), [292](#), [293](#), [296](#), [304](#)

## 3NC

three-Noun Compound. [4](#), [5](#), [29](#), [34](#), [35](#), [42](#), [44](#), [45](#), [50](#), [69](#), [72](#), [85](#), [99](#), [139](#), [144–146](#), [158](#), [233–236](#), [264](#), [266](#), [275](#), [277](#), [281](#), [283](#), [287–293](#), [297](#), [299](#), [301](#), [306](#), [307](#), [312–314](#), [316](#), [317](#), [320–325](#), [327](#), [338–342](#), [344–349](#), [352–354](#), [357](#), [358](#), [362](#), [369](#), [370](#), [395](#), [397](#), [408](#), [414](#)

## 4NC

four-noun Compound. [69](#), [71](#), [72](#), [196](#), [277](#), [281](#), [314](#), [317–319](#), [326](#), [329](#), [332](#), [333](#), [337–342](#), [344](#), [345](#), [349](#), [351](#), [354](#), [397](#)

## 5NC

five-noun Compound. [319](#), [355](#)

## A

### AdjDepMod

Adjacency-Dependency Model. [299](#), [355](#), [356](#), [390](#)

**AdjMod**

adjacency model. 282–285, 287, 289–292, 298, 299, 301, 311, 322, 355, 356

**AM**

Association Measure. 87, 88, 91–94, 282–285, 291, 292, 298, 299, 301, 302, 341, 343, 347, 398, 405

**APP**

Aligned Phrase Pattern. 22, 145, 146, 280, 282, 285, 306–315, 325, 344, 346, 347, 349–351, 353, 362, 390, 397

**APPP**

Aligned Phrase Pattern Parsing. 282, 291, 309–311, 313, 315, 322–324, 344, 346–349, 355, 362, 369, 397

**APPP<sub>WA</sub>**

aligned phrase pattern parsing supported by word alignment. 285, 311, 313, 315, 344, 397

**AWD**

aligned word distance. 22, 139, 279, 280, 282, 285, 291, 292, 315–320, 323, 325–335, 337, 341, 344–348, 351–356, 362, 392, 405, 411

**AWS**

aligned word set. 139, 315–319, 352, 399

**B**

**BC**

binary compound. 29

**BLEU**

BiLingual Evaluation Understudy. 166, 177, 179, 246

**C**

**CA**

Conceptual Association. 288

**CCR**

Closed Compound Restrictor. [140–144](#), [147–149](#), [151](#), [153](#), [154](#), [312](#), [320](#), [338](#), [364](#), [395](#)

**CRF**

Conditional Random Fields. [89](#)

**D**

**DBUP**

deterministic bottom-up parsing. [22](#), [317–329](#), [336](#), [338](#), [340–342](#), [344](#), [345](#), [348](#), [362](#), [365](#), [397](#)

**DC**

Determination of [Compoundhood](#). [4](#)

**DepMod**

dependency model. [282–285](#), [287](#), [289–292](#), [294](#), [295](#), [298](#), [299](#), [311](#), [322](#), [355](#), [357](#)

**DS**

Distributional Semantics. [13](#), [165](#), [170–172](#), [225](#), [231](#), [234](#), [257](#), [271](#), [300](#)

**Dsim**

Distributional Similarity. [21](#), [161–164](#), [172](#), [175](#), [176](#), [225–231](#), [234](#), [236](#), [237](#), [257](#), [258](#), [263](#), [265](#), [266](#), [269](#), [271](#), [362](#), [396](#)

**DSM**

Distributional Semantics Model. [172](#), [175](#), [225](#), [231](#), [232](#), [258](#), [271](#)

**DT**

Distributional Thesaurus. [172](#)

**E**

**ED**

Edit Distance. [169](#), [180](#), [182](#), [188](#), [189](#), [256](#), [372](#), [373](#), [375](#), [377](#)

**ENCD**

Europarl Nominal Compound Database. 19, 21, 79, 84, 86, 90, 96–99, 107, 138, 140, 141, 145–149, 151, 154, 299, 307, 312, 318, 320, 321, 338, 341, 344, 347, 349, 357, 362, 379, 406

**ENCR**

Europarl Nominal Compoundhood Ratings. 19, 84, 96, 97, 116, 120, 123, 147–151, 154

**F**

**FTA**

Full parse Tree Accumulation. 329, 331–333, 337, 397

**I**

**IAA**

Inter-Annotator Agreement. 13, 80, 82, 107, 119–124, 130, 131, 147, 148, 150–152, 312, 321, 341, 364

**IE**

Information Extraction. 11, 240

**IR**

Information Retrieval. 11, 156, 167, 172, 240, 339, 340, 345, 349

**K**

**kC**

*k*-ary compound. 29, 45, 68, 71, 72, 320, 333

**kNC**

*k*-noun Compound. 29, 45, 281, 338, 340, 341, 349–351, 369, 399, 414

**L**

**LC**

linguistic criterion. 115–120, 123–125, 127–132, 153

*List of Abbreviations*

**LCI**

Linguistic Criterion Inspection. [4](#), [76](#), [82](#), [84](#), [115](#), [117](#), [130](#), [131](#), [136](#), [150](#), [151](#), [153](#), [362](#), [363](#)

**LR**

Lemma Resource. [185–188](#), [191](#), [192](#), [194](#), [199](#), [204](#), [269](#)

**LSF**

lemma sequence format. [196](#), [197](#), [199–201](#), [203–206](#), [210](#), [211](#), [227–232](#), [242](#), [256](#), [271](#), [372](#), [373](#), [375](#), [378](#), [409](#)

**M**

**ME**

Maximum Entropy. [173](#), [299](#), [303](#)

**MI**

Mutual Information. [91](#), [92](#), [166](#), [172](#), [287](#), [288](#), [291](#)

**MOP**

Morphological Operation Pattern. [21](#), [160](#), [161](#), [163](#), [180–184](#), [187](#), [188](#), [190–194](#), [198](#), [199](#), [201](#), [202](#), [207](#), [210–220](#), [233](#), [253](#), [256](#), [257](#), [259–263](#), [265](#), [269](#), [270](#), [362](#), [373](#), [374](#), [376](#), [377](#), [391](#), [396](#), [407](#), [410–412](#), [419](#)

**MR**

MOP Resource. [187](#), [188](#), [192](#)

**MS**

MOP Suitability. [188](#), [189](#), [256](#)

**MT**

Machine Translation. [10](#), [15](#), [19](#), [195](#), [240](#), [246](#), [276](#), [290](#)

**MWE**

Multi-Word Expression. [24](#), [25](#), [27](#), [29](#), [66](#), [75](#), [85](#), [87](#), [90–95](#), [106–110](#), [137](#), [349](#), [401](#), [403](#), [416](#)

**N**

**NC**

Noun Compound. [30](#), [292](#)

**NER**

Named Entity Recognition. [89](#), [108](#), [299](#), [300](#)

**NFTAP**

non-deterministic full tree accumulation parsing. [22](#), [326](#), [328](#), [329](#), [333](#), [335–337](#), [340–342](#), [344](#), [345](#), [356](#), [362](#)

**NLG**

Natural Language Generation. [81](#)

**NLP**

Natural Language Processing. [2](#), [3](#), [7](#), [10–13](#), [15](#), [17](#), [43](#), [45](#), [56](#), [77](#), [80–82](#), [88](#), [97](#), [110](#), [111](#), [113](#), [145](#), [156](#), [163](#), [166](#), [195](#), [196](#), [198](#), [222](#), [230](#), [240](#), [245](#), [258](#), [263](#), [275](#), [276](#), [285](#), [290](#), [293](#), [391](#), [405](#)

**NLU**

Natural Language Understanding. [2](#), [3](#), [10](#), [80](#), [156](#), [246](#), [276](#)

**NP**

noun phrase. [30](#), [43–45](#), [95](#), [101](#), [106](#), [116](#), [139](#), [141](#), [143](#), [144](#), [152](#), [275](#), [276](#), [279](#), [281](#), [294](#), [295](#), [297–302](#), [412](#)

**NSTAP**

Non-deterministic Subtree Accumulation Parsing. [22](#), [327](#), [335–337](#), [340–342](#), [345](#), [354](#), [363](#)

**NTAP**

Non-deterministic Tree Accumulation Parsing. [326](#), [327](#), [332](#), [339](#), [341](#), [342](#), [344](#), [345](#), [348](#), [349](#)

**O**

**OOV**

Out-Of-Vocabulary. [167](#), [177](#)

**P**

**PCFG**

Probabilistic Context-Free Grammar. [303](#), [304](#)

**PMI**

Pointwise Mutual Information. [92](#), [94](#), [294](#), [300–302](#)

**PoS**

Part-of-Speech. [13](#), [19](#), [29](#), [32](#), [33](#), [46–48](#), [50](#), [51](#), [53](#), [84](#), [85](#), [87–89](#), [94](#), [95](#), [98](#), [104](#), [105](#), [109](#), [112](#), [113](#), [122](#), [134](#), [137](#), [138](#), [141](#), [142](#), [154](#), [168](#), [170](#), [174](#), [176](#), [185](#), [189](#), [192](#), [194](#), [198](#), [199](#), [203](#), [210–216](#), [220](#), [221](#), [233](#), [247](#), [248](#), [250](#), [251](#), [256](#), [262](#), [263](#), [270](#), [293](#), [299](#), [301](#), [303](#), [312](#), [339](#), [366](#), [369](#), [370](#), [394](#), [413](#)

**PP**

Prepositional Phrase. [10](#), [34](#), [43](#), [44](#), [91](#), [288](#), [290](#), [303](#), [343](#)

**PTB**

Penn Treebank. [276](#), [297–300](#)

**Q**

**QA**

Question Answering. [11](#), [240](#), [250](#), [273](#)

**R**

**RHHR**

RightHand Head Rule. [33](#), [189](#)

**RQ**

Research Question. [15](#), [16](#), [82](#), [83](#), [151–153](#), [161–163](#), [259–268](#), [278–280](#), [345–348](#), [363](#), [364](#)

**RTE**

Recognizing Textual Entailment. [10](#), [11](#), [21](#), [156](#), [162](#), [163](#), [165](#), [166](#), [222](#), [239–246](#), [248–254](#), [258](#), [259](#), [267–269](#), [272](#), [273](#), [362](#), [397](#), [407](#)

**S**

**SA**

[structural analysis](#). [5](#)

**SemRel**

[semantic relation](#). [293](#)

**SiMode**

[Similarity Mode](#). [228](#), [229](#), [232](#), [234–238](#), [258](#), [263](#), [266](#), [267](#), [271](#)

**SMT**

[Statistical Machine Translation](#). [80](#), [81](#), [92](#), [111](#), [156](#), [157](#), [159](#), [160](#), [162](#), [163](#), [166–169](#), [171](#), [173–179](#), [203](#), [239](#), [241](#), [245](#), [246](#), [258](#), [267](#), [268](#), [272](#), [406](#)

**SPF**

[split point format](#). [22](#), [186](#), [191](#), [196](#), [197](#), [199–201](#), [203–206](#), [210](#), [211](#), [222](#), [256](#), [271](#), [372–378](#), [390](#), [415](#)

**SR**

[Speech Recognition](#). [156](#), [167](#), [176](#)

**STA**

[SubTree Accumulation](#). [335](#), [336](#)

**SVM**

[Support Vector Machine](#). [73](#), [172](#), [293](#), [294](#), [300](#), [301](#)

**T**

**TC**

[ternary compound](#). [29](#), [44](#), [283–285](#), [306](#), [308](#), [313](#), [392](#)



*List of Abbreviations*

**TE**

Textual Entailment. [10](#), [239–241](#), [247](#), [258](#), [267](#), [397](#)

**TER**

Translation Edit Rate. [177](#)

**TTS**

Text-to-Speech. [10](#), [50](#)

**U**

**USP**

universal surface pattern. [22](#), [132–135](#), [144–146](#), [152](#), [306](#), [307](#), [349](#), [369](#), [370](#), [395](#), [397](#), [417](#)

**W**

**wFST**

weighted Finite State Transducer. [172](#), [174](#)

**WSD**

Word Sense Disambiguation. [8](#), [367](#), [379](#)

**X**

**XCI**

Cross-lingual Compound Inspection. [4](#), [65](#), [83](#), [84](#), [132–135](#), [140](#), [151](#), [152](#), [154](#), [280](#), [306](#), [307](#), [362](#), [363](#), [395](#)

*List of Abbreviations*

# List of Algorithms

18.1 MOP application-based lemma lookup . . . . .	203
24.1 APPP with cross-lingual head correlation assumption . . . . .	329
24.2 APPP with word alignment interpretation . . . . .	330
25.1 Deterministic Bottom-Up Parsing . . . . .	338
25.2 Non-deterministic Full Tree Accumulation Parsing . . . . .	348
25.3 Non-deterministic Subtree Accumulation Parsing . . . . .	355
26.1 Bootstrapping of APPs for a given compound size and structure $\Sigma$ . . . . .	370
26.2 AdjDepMod Annotation Function . . . . .	375
26.3 Non-parallel approach to cross-lingually parsing a compound <b>A B C</b> . . . . .	379
C.1 Linear SPF compilation . . . . .	397

*List of Algorithms*

# List of Figures

1.1	Compound Analysis in NLP . . . . .	4
1.2	Thematic structure of the thesis . . . . .	5
1.3	Linear splitting for <i>Hühnersuppenrezept</i> . . . . .	6
1.4	Split tree for <i>Hühnersuppenrezept</i> . . . . .	7
1.5	Ambiguity in various Compound Analysis Levels . . . . .	10
3.1	Example of a compound tree structure . . . . .	39
3.2	Tree structure for plastic water bottle . . . . .	39
3.3	Compound taxonomy by Bisetto and Scalise (2005) . . . . .	43
3.4	Tree structure for ‘Natural Language Processing’ . . . . .	47
3.5	Structural ambiguity in German with different primary stress . . . . .	54
5.1	Example of a balanced tree structure . . . . .	78
8.1	Sample data in Nicholson and Baldwin (2008) . . . . .	111
8.2	Example sentence annotation in the Wiki50 corpus (Vincze et al., 2011) . . . . .	117
8.3	Example annotations in the CMWE corpus (Schneider et al., 2014b) . . . . .	118
10.1	OpenOffice spreadsheet for nominal compound annotation . . . . .	125
10.2	Examples of spreadsheet-based nominal compound annotation . . . . .	126
10.3	Distribution of Compoundhood and LC Ratings . . . . .	134
10.4	The J48 decision tree for all linguistic criteria . . . . .	137
11.1	Cross-lingual Compound Identification Method . . . . .	147
15.1	Structures for the German <i>Hochschulgebäude</i> ‘university building’ . . . . .	173
16.1	Example of a split lattice in Dyer (2009) . . . . .	188
18.1	Architecture of the splitting algorithm . . . . .	200
18.2	Constituent normalization by using MOP application . . . . .	202

*List of Figures*

18.3	Example of a split tree structure with related MOPs . . . . .	205
18.4	False compound split of <i>Quartal</i> ‘quarter (year)’ . . . . .	207
18.5	False compound split of <i>Läuferteam</i> ‘runner’s team’ . . . . .	209
18.6	Linguistically motivated split of <i>Schauspielzeitschrift</i> . . . . .	210
18.7	Tree pruning for <i>Schauspielzeitschrift</i> . . . . .	211
18.8	Example of a compound split in the evaluation format . . . . .	215
18.9	Examples of a ranking output produced by the splitting system of Weller and Heid (2012) . . . . .	226
20.1	Dataflow proposed by Noh et al. (2015) . . . . .	263
21.1	Example of a ternary split tree structure for <i>Langsamfahrstelle</i> . . . . .	286
22.1	Example of binary structure for sentence parsing . . . . .	293
22.2	Tree structures for ‘Natural Language Processing’ . . . . .	294
22.3	Balanced tree structure for <i>energy efficiency action plans</i> . . . . .	296
23.1	Dependency relations for LEFT- and RIGHT-branched <i>natural language processing</i> . . . . .	302
23.2	Dependency relations for swapped modifiers in a RIGHT-branched TC . .	302
23.3	Parse tree for <i>wooden French onion soup bowl handle</i> . . . . .	315
25.1	DBUP parse tree for <i>air transport safety organization</i> . . . . .	339
25.2	DBUP parse tree for <i>twin pipe undersea gas pipeline</i> . . . . .	340
25.3	Partial result for parsing <i>book price fixing schemes</i> using a Danish phrase	346
25.4	Possible parse trees for <i>air traffic control</i> . . . . .	347
25.5	Possible binary parse trees for <i>energy efficiency action plan</i> . . . . .	351
25.6	Semantically equivalent trees for <i>farm income stabilisation instrument</i> . .	352
25.7	Possible binary parse trees for <i>church development aid projects</i> . . . . .	354
25.8	Two possible parse trees for <i>church development aid projects</i> annotated with AWDs in Italian, German and French . . . . .	357
26.1	Trees for <i>world reserve currency</i> . . . . .	374
26.2	Tree structure for <i>twin pipe undersea gas pipeline</i> . . . . .	376
26.3	AdjDepMod-annotated parse trees for <i>world reserve currency</i> . . . . .	376
27.1	Thematic structure of the thesis . . . . .	383
27.2	Compound Analysis in NLP . . . . .	387

*List of Figures*

27.3 Ambiguity in various Compound Analysis Levels . . . . . 388

C.1 Example of a split tree with recursive lemma splitting . . . . . 399

E.1 OpenOffice spreadsheet for nominal compound annotation . . . . . 411

E.2 Examples of spreadsheet-based nominal compound annotation . . . . . 412

*List of Figures*



# List of Publications

Parts of the research described in this thesis have been published in:

## 2014

- Ziering, P. and Van der Plas, L. (2014). What good are 'Nominalkomposita' for 'noun compounds': Multilingual Extraction and Structure Analysis of Nominal Compositions using Linguistic Restrictors. In *COLING 2014*  
⇒ presented and elaborated in Part B, Part C and in Part E, Chapter 24

## 2015

- Ziering, P. and Van der Plas, L. (2015a). From a Distance: Using Cross-lingual Word Alignments for Noun Compound Bracketing. In *IWCS 2015*  
⇒ presented and elaborated in Part E, Chapter 25
- Ziering, P. and Van der Plas, L. (2015b). One tree is not enough: Cross-lingual accumulative structure transfer for semantic indeterminacy. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 739–746, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA  
⇒ presented and elaborated in Part E, Chapter 25

## 2016

- Ziering, P. and Van der Plas, L. (2016). Towards Unsupervised and Language-independent Compound Splitting using Inflectional Morphological Transformations. In *NAACL-HLT 2016*  
⇒ presented and elaborated in Part D, Chapter 17 and Chapter 18
- Ziering, P., Müller, S., and Van der Plas, L. (2016). Top a Splitter: Using Distributional Semantics for Improving Compound Splitting. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 50–55, Berlin, Germany. Association for Computational Linguistics  
⇒ presented and elaborated in Part D, Chapter 19

**2017**

- Jagfeld, G., Ziering, P., and Van der Plas, L. (2017). Evaluating Compound Splitters Extrinsically with Textual Entailment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada. Association for Computational Linguistics  
⇒ presented and elaborated in Part D, Chapter 20

# List of Tables

3.1	Recursive compound construction for <i>orange juice</i> . . . . .	38
3.2	Number of possible binary trees for compounds with $k$ constituents . . . . .	40
3.3	Possible PoS combinations for English binary compounds . . . . .	50
3.4	Possible PoS combinations for German binary compounds . . . . .	52
3.5	Possible PoS combinations for Dutch binary compounds . . . . .	55
3.6	Distribution of linking elements in Afrikaans compounding, by Huyssteen and Zaanen (2004) . . . . .	58
3.7	Distribution of compound size in Afrikaans, by Huyssteen and Zaanen (2004) . . . . .	59
5.1	Examples of parallel compounding . . . . .	71
5.2	Examples of translations of adjective-noun sequences . . . . .	72
5.3	Ratios of parallel compounding . . . . .	72
5.4	Ratios of parallel compounding for adjective-noun phrases . . . . .	72
5.5	Examples for parallel closed compounding . . . . .	73
5.6	Examples of phrasal equivalents in Romance languages . . . . .	74
5.7	Ratios of phrasal equivalents in Romance languages . . . . .	74
5.8	Cross-lingual modifier alternations . . . . .	75
5.9	Examples of French phrasal equivalents for <i>made_of</i> compounds . . . . .	79
8.1	Languages in MORBO/COMP . . . . .	106
8.2	Information fields in MORBO/COMP . . . . .	107
8.3	Semantic relation class frequency distribution in Ó Séaghdha (2007) . . . . .	108
8.4	Distribution of verbal classes in Nicholson and Baldwin (2008) . . . . .	111
8.5	Examples of compositionality ratings by Reddy et al. (2011) . . . . .	112
8.6	Meaningfulness ratings in Graves et al. (2013) . . . . .	113
8.7	Examples of 2NCs in the dataset of Farahmand et al. (2015) . . . . .	114
8.8	MWE distribution in the Wiki50 corpus . . . . .	116

List of Tables

9.1	EUROPARL language selection . . . . .	120
10.1	Size of the final ENCR datasets . . . . .	127
10.2	Agreement on compound extraction in the <i>training stage</i> . . . . .	129
10.3	Agreement on compound extraction in the <i>agreement set</i> . . . . .	129
10.4	Exclusive extractions for both annotators . . . . .	130
10.5	Average difference in compoundhood rating in the <i>training stage</i> . . . . .	130
10.6	Average difference in compoundhood rating in the <i>agreement set</i> . . . . .	131
10.7	Average difference in LC rating in the <i>training stage</i> (1) . . . . .	131
10.8	Agreement on LC rating in the <i>training stage</i> (2) . . . . .	132
10.9	Agreement on LC rating in the <i>agreement set</i> . . . . .	133
10.10	Compoundhood prediction accuracy using a J48 decision tree . . . . .	135
10.11	Compoundhood prediction using feature groups in a J48 decision tree . . . . .	136
10.12	Statistical significance test for a J48 decision tree . . . . .	136
10.13	Compoundhood prediction accuracy using a Naive Bayes classifier . . . . .	138
10.14	Compoundhood prediction using feature groups in a Naive Bayes classifier . . . . .	138
10.15	Statistical significance test for a Naive Bayes Classification . . . . .	138
10.16	Average difference between compoundhood and linguistic criteria . . . . .	139
10.17	XCI - USP frequency distribution for all aligned languages . . . . .	143
10.18	XCI - USP frequency distribution for Germanic languages . . . . .	143
10.19	XCI - USP frequency distribution for Greek . . . . .	144
10.20	XCI - USP frequency distribution for Romance languages . . . . .	144
11.1	Predefined PoS patterns for the compound candidate selection . . . . .	148
12.1	PoS pattern distribution in $CCR(n)$ . . . . .	154
12.2	The 10 most frequent English JJ NN sequences in $CCR(0)$ and $CCR(4)$ . . . . .	155
12.3	The 10 most frequent English NN POS NN sequences in $CCR(0)$ and $CCR(4)$ . . . . .	155
12.4	Degree of closed compounding among the closed compounding languages in $CCR(1)$ . . . . .	156
12.5	The 10 most frequent equivalents of English 3NCs in USP format . . . . .	157
13.1	Compound Identification Results for <b>Annotation1</b> dataset . . . . .	160
13.2	Compound Identification Results for <b>Annotation2</b> dataset . . . . .	161
13.3	Compound Identification Results for <b>Combination</b> dataset . . . . .	161
16.1	Splitting example from Larson et al. (2000) . . . . .	182

*List of Tables*

16.2	Example of undersplitting in Koehn and Knight (2003)	183
16.3	Modifications proposed by Stymne (2008)	183
16.4	SMT performance for different compound splitters	192
16.5	Intrinsic performance for different compound splitters	193
17.1	Examples of MOPs for <i>German, Dutch and Afrikaans</i>	196
18.1	Word MOPs for the lemma <i>Termin</i> ‘appointment’	206
18.2	Word MOPs for the lemma <i>Lauf</i> ‘run’	208
18.3	Training data statistics for multilingual compound splitting	214
18.4	Overlap between German Word MOPs and Gold-constituent MOPs in HH2011GS	216
18.5	Distribution of the number of gold constituents used in M2006GS	218
18.6	Splitting gold standard overlap between HH2011GS and M2006GS	220
18.7	Distribution of the number of gold constituents used in HB2008GS	220
18.8	Splitting gold standard overlap between HH2011GS+M2006GS and HB2008GS	221
18.9	Distribution of the number of gold constituents used in VZDH2014GS	222
18.10	Overlap between German word MOPs and gold-constituent MOPs in VZDH2014GS/NL	222
18.11	Overlap between German word MOPs and gold-constituent MOPs in VZDH2014GS/AF	222
18.12	Evaluation of all proposed compound splitting features; numbers in %	228
18.13	Results for German compound splitting - MOP set comparison; numbers in %	230
18.14	Results for Dutch and Afrikaans compound splitting - MOP set compar- ison; numbers in %	232
18.15	Results for German compound splitting - external system comparison	235
18.16	Distribution of splitting depths in M2006GS	236
18.17	Examples of different splitting depths for ZvdP2016 and WH2012	237
18.18	Results for Dutch/Afrikaans compound splitting - external system com- parison	237
19.1	Initial split ranking	242
19.2	Distributional similarity between compound and constituents	243
19.3	Geometric mean Dsim scores	243
19.4	Split re-ranking with GEO scores	244

List of Tables

19.5	Test set coverage for compound split re-ranking . . . . .	248
19.6	Results of split re-ranking for FF2010 . . . . .	251
19.7	Results of split re-ranking for WH2012 . . . . .	252
19.8	Results of split re-ranking for ZvdP2016 . . . . .	253
20.1	Examples for Textual Entailment (TE) . . . . .	256
20.2	Results on RTE performance with prior compound splitting . . . . .	268
23.1	Structure class distribution in Lauer (1994) . . . . .	306
24.1	Description of USP tags . . . . .	326
24.2	The 10 most frequent paraphrases of English 3NCs in USP format . . . . .	326
24.3	Six APPs and the corresponding structure . . . . .	327
24.4	Examples of paraphrases for the six selected APPs . . . . .	328
24.5	Parsing results for APPP and APPP <sub>WA</sub> . . . . .	332
25.1	Parsing coverage for DBUP and systems in comparison . . . . .	343
25.2	Parsing results for DBUP and systems in comparison on common test subsets . . . . .	344
25.3	Comparison of different language families for type-based DBUP . . . . .	344
25.4	Number of possible binary parse trees for compounds with $k$ constituents . . . . .	349
25.5	FTA for the semantically indeterminate <i>farm income stabilisation instrument</i> . . . . .	353
25.6	Frequency distribution of bracketing patterns in the 4NC test set . . . . .	359
25.7	Parsing results in MRP for 3NCs and 4NCs . . . . .	362
25.8	Parsing results for 4NCs . . . . .	362
A.1	Mapping from words to Universal Surface Patterns . . . . .	394
B.1	German constituent inflection operations (Langer, 1998) . . . . .	395
C.1	Evaluation of the linear split point format compilation for FF2010 . . . . .	400
C.2	Evaluation of the different split point format compilers . . . . .	402
D.1	Splitting gold standard overlap between HH2011GS+M2006GS+HB2008GS and Ghost-NN . . . . .	406

# List of Terms

[Symbols](#) | [A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#)

## Symbols

### *k*-ary compound

A [compound](#) with *k* ([atomic](#)) [constituents](#), i.e., with a [compound size](#) of *k*. [29](#), [45](#), [137](#), [384](#)

## A

### adjacency model

A [parsing](#) model for [ternary compounds](#) (A B C), where the association of adjacent [constituents](#) are compared, i.e., [AM](#)(A,B) vs. [AM](#)(B,C). It was initially proposed by Marcus (1980). [282–284](#), [287](#), [298](#), [311](#), [333](#), [343](#), [348](#), [354](#), [357](#), [382](#)

### adjectival compound

A [compound](#) with an adjective as [head](#) (e.g., *bulletproof*). [29](#), [37](#), [46–48](#), [51](#), [85](#), [97](#), [107](#), [108](#), [153](#), [366](#)

### adjective compound

A [compound](#) composed of only adjectives (e.g., *dark-blue* but not *bulletproof*). [29](#), [37](#), [51](#)

### aligned phrase

A paraphrase (consisting of several [content words](#) usually separated by [function words](#)) that is [cross-lingually](#) aligned to a [target compound](#) (i.e., a [phrasal equivalent](#)). [277](#), [280](#), [344](#), [347](#), [354](#), [413](#)

### aligned word distance

The minimum distance between of the [cross-lingual equivalents](#) of two [target constituents](#) (i.e., of the [constituent equivalents](#)) in a translated sentence. A formal definition is given in Section 25.1. [139](#), [279](#), [280](#), [282](#), [291](#), [315](#), [344](#), [351](#), [362](#), [382](#), [405](#)

### aligned word set

The set of [cross-lingually](#) aligned words of a [target constituent](#)  $c_i$ ,  $AWS(c_i)$ , i.e., the set of (possibly discontinuous) [constituent equivalents](#). A formal definition is given in Section 25.1. [139](#), [315](#), [382](#), [403](#)

### allomorph

A morpho-phonologically adapted variant of a [lemma](#), e.g., the Dutch *boll* for the [lemma](#) *bol* ‘bulb’ (as in *bloembollenveld* ‘flower bulb field’). [173](#), [206](#), [218](#), [219](#)

### atomic

An **atomic** term is simplex and indivisible with respect to [compounding](#), i.e., the counterpart of [compounds](#). While derivations are considered to be **atomic**, any [compound](#) is complex. During [compound splitting](#), a [compound](#) can be split recursively into [constituents](#) as long as these [constituents](#) are not predicted to be **atomic**. [2](#), [5–8](#), [14](#), [15](#), [17](#), [18](#), [24](#), [26](#), [29](#), [32](#), [42](#), [55](#), [70](#), [76](#), [83](#), [84](#), [113](#), [133](#), [140](#), [143](#), [167](#), [171](#), [184](#), [185](#), [191](#), [192](#), [194](#), [195](#), [200](#), [205](#), [219–222](#), [227](#), [231](#), [233](#), [240–243](#), [245](#), [247](#), [256](#), [264](#), [268](#), [272](#), [276](#), [277](#), [283](#), [284](#), [294](#), [296](#), [300](#), [315–318](#), [320](#), [325](#), [344](#), [346](#), [347](#), [379](#), [398](#), [399](#), [401](#), [402](#), [405](#), [408](#), [411](#)

## B

### base noun phrase

A [base noun phrase](#) is a [noun phrase](#) composed of a nominal [head](#) and some prenominal [modifiers](#), e.g., adjectives or (in the case of a [kNC](#)) nouns. [Base noun phrases](#) do not contain postnominal attributes such as relative clauses. “Base NPs also contain determiners, possessives, adjectives, and conjunctions” (Pitler et al., 2010). [276](#), [294](#), [297–300](#), [303](#), [343](#), [399](#), [412](#), [413](#)

### binary compound

A [compound](#) with exactly two [atomic constituents](#) (e.g., *data<sub>1</sub> base<sub>2</sub>* or *blue<sub>1</sub>-eyed<sub>2</sub>*). [17](#), [18](#), [29](#), [46](#), [48](#), [51](#), [55](#), [61](#), [68](#), [82](#), [99](#), [101](#), [103](#), [105](#), [106](#), [137](#), [184–186](#), [189–192](#),



194, 195, 200–202, 206, 216, 217, 221, 228, 232, 233, 239, 256, 264, 266, 269, 271, 275, 313, 328, 329, 335, 338, 341, 344, 345, 382, 394, 397, 411, 413

### **bracketing**

This term is sortally polysemous. While previous work often use ‘bracketing’ for referring to the method of producing a **bracketing** structure, we call this method ‘parsing’ and use the term ‘bracketing’ as a representation of the **parsing** result (as an alternative to a **parse tree**), where **constituents** are grouped using square brackets. 5, 35, 44, 158, 172, 186, 199, 200, 203, 233–237, 266, 287, 292–294, 297, 300, 333, 353, 400, 402

### **bracketing pattern**

A **bracketing** representation in which the **constituents** are generalized to capitalized letters starting with A, e.g., ‘[A B][C D]’ for the **bracketing** [*air traffic*][*control center*]. 338, 339, 341, 397

## **C**

### **closed compound**

A one-word **compound**, i.e., a **compound** spelled with a single **word**. In this thesis, we count **hyphenated compounds** as **closed compounds** (e.g., *part-time*, *timetable* or the German *Computerlinguistik* ‘computational linguistics’). 3, 4, 8, 12, 14, 17, 18, 24, 25, 29, 31, 32, 34, 46, 47, 55, 59, 60, 64, 66–68, 71, 75, 80–83, 85–87, 90, 92, 95–97, 106–108, 113, 116, 133, 135, 137, 140, 141, 143, 144, 147–150, 153, 154, 156, 157, 160, 165, 167, 175, 234, 241–245, 251, 258, 268, 269, 277, 281, 293, 306, 308, 312, 316–318, 323, 325, 326, 346, 353, 363, 364, 369, 400–402, 414, 415, 418

### **closed compounding**

A language’s property of creating **closed compounds**. 10, 34, 55, 59, 60, 67, 71, 96, 135, 140, 143–145, 147, 153, 154, 157, 165, 277, 291, 367, 394, 395, 400

### **closed compounding language**

A language with **closed compounding**, i.e., in which **compounds** are usually **closed**, such as the Germanic languages *German*, *Dutch*, *Danish*, *Swedish* or *Afrikaans*. 2, 31, 32, 47, 50, 52, 53, 55, 66–68, 83, 86, 94, 106, 112, 131, 133, 135, 140, 141, 143, 144, 147, 152, 153, 156, 165, 171, 182, 193, 197, 231, 250, 256, 263, 273, 312, 325, 395

### complex lexeme

A **complex lexeme** is a **lexeme** that is composed of several **atomic lexemes**. Besides **Multi-Word Expressions (MWEs)**, **complex lexemes** also include complex single words, i.e., **closed compounds**. 2, 6, 10, 24, 28, 51, 401

### complex nominal

A nominal **MWE** including a preposition or other **functional** markers between the nominal **constituents**. Complex nominals are often found in Romance languages, e.g., the Italian *succo di limone* ‘lemon juice’ or *porta a vetri* ‘glass door’ (Baldwin and Kim, 2010). We also consider similar constructions in English (e.g., *part of speech* or *hall of fame*) as **complex nominals**. 10, 14, 18, 28, 29, 34, 68, 70, 73, 91, 100, 101, 133–135, 152, 156, 308, 325, 346, 351, 401

### compound

A **compound** is the “formation of a new **lexeme** by adjoining two or more **lexemes**” (Bauer, 2003). However, the clear definition and even the existence of **compounds** is controversially discussed in linguistics literature (Lieber and Štekauer, 2009). More details about **compounds** are discussed in Part B. 2–12, 14–20, 24–42, 44, 46–77, 79–88, 90, 91, 95–99, 101–107, 110–112, 115–124, 127, 130–132, 135–138, 140, 144–154, 158–160, 164–177, 181, 182, 184–186, 189, 190, 193–197, 200–203, 205, 206, 209–218, 220–224, 226–238, 241–246, 253, 257–261, 265–272, 275–279, 281–283, 285, 287, 288, 291, 292, 294, 297, 310, 312, 314, 315, 320–323, 328, 329, 333, 335, 339, 344–348, 357, 358, 361–363, 365–367, 369, 373, 375, 377, 379, 380, 394–408, 410–418

### compound analysis

The automatic **analysis of compounds**, starting with the determination of **compoundhood**, followed by the **structural analysis** and finally the semantic analysis, as illustrated in Figure 1.1. 7–10, 12–16, 18, 19, 25, 26, 65, 69, 70, 76, 77, 80–82, 84, 161, 361, 364, 366, 369, 401

### compound class

The **class** of a **compound** according to a predefined taxonomy (e.g., that of Bisetto and Scalise (2005)). Possible **classes** are **endocentric** subordinate **compounds** (e.g., *sun glasses*) or **exocentric** coordinate **compounds** (e.g., *north east*). A more detailed presentation of a **compound** taxonomy is given in Section 3.7. 8, 168, 401

### **compound equivalent**

A **cross-lingual equivalent** being a **compound**, e.g., a **cross-lingually aligned closed compound**. 151, 332, 363, 365, 405

### **compound parsing**

The task of structuring a **compound** with three or more **constituents**, i.e., revealing the **internal structure** and producing a **parse tree** or a **bracketing**. 5, 11, 14, 17–22, 26, 36, 45, 65, 71–73, 76, 77, 81, 113, 145, 146, 234, 264, 275–283, 285–303, 306, 310–315, 317, 318, 320, 322–326, 333, 336, 338, 341–349, 351, 353–355, 357, 358, 361, 362, 364–367, 369, 370, 398, 400, 405, 409, 411, 413, 415, 416

### **compound size**

The number of **atomic constituents**,  $k$ , a **compound** is composed of. 18, 19, 29, 35, 36, 55, 84, 137, 203, 205, 207, 220, 294, 301, 320, 329, 347, 349, 350, 356, 390, 394, 398

### **compound splitting**

A **decompounding** task where a **closed compound** is dissected into the composed **lexemes**, i.e., its (**atomic**) **constituents** (see Part D). 4, 10–12, 14, 17–22, 26, 31, 65, 71, 76, 77, 81, 90, 96, 97, 106, 107, 113, 156–179, 181, 183–186, 189–206, 208–224, 226–246, 248–273, 278, 281, 290, 361, 362, 364, 366, 367, 372, 373, 375, 376, 378–380, 391, 396, 397, 399, 402, 406, 407, 409, 411, 412, 414–417

### **compoundhood**

The property of being a **compound**. 3, 4, 7, 11, 15, 17–21, 24, 60, 64, 75, 76, 79, 82–84, 97, 115–118, 120, 122–124, 127, 129–132, 147, 148, 150–154, 301, 361–363, 366, 367, 383, 401, 410

### **compoundhood status**

The decision whether a target **term**  $\Psi$  is a **compound** or not. 3, 7, 11, 13–16, 19, 80, 81, 115, 131, 137, 150, 379

### **compounding**

The process of creating **compounds**. 6, 9, 10, 14, 15, 17, 19–21, 24, 25, 29, 31, 46, 49, 50, 52, 54–56, 65–68, 71, 75–77, 79, 81, 83, 86, 87, 96, 111, 112, 117, 132, 137, 143, 146, 156, 160, 165, 174, 180, 188, 251, 252, 254, 259, 260, 277, 362, 379, 394, 399, 403

### constituent

A **constituent** is a unit of one or more **words** being part of a **compound** or **MWE**. The inflected form of a **constituent** (e.g., including **linking elements**) is called **constituent form**, whereas the **normalized** version is called **constituent lemma**. There are mediate and **immediate constituents**. 3, 5–8, 10, 12, 15, 18, 24, 25, 28–38, 40–42, 44, 47, 49, 50, 53–56, 58, 59, 61, 69–71, 75, 84, 88, 91, 98, 99, 101, 104–107, 109, 113, 118, 131, 135, 137–139, 148, 154, 158–161, 163, 164, 167–176, 184–187, 189, 191, 194–197, 200–207, 210–214, 216–221, 223, 224, 226–228, 230–232, 234, 241–245, 253, 257, 260–265, 268–272, 275–277, 279, 281, 283, 285, 291–296, 300, 301, 310–312, 314–318, 320, 325–329, 333, 335, 339, 344, 346–348, 355, 357, 363, 366, 367, 370, 373, 378–380, 391, 394, 396–405, 407, 408, 410, 412, 414–417

### constituent alphabet

This is the language-specific set of characters that are allowed within a **constituent**. In particular, characters indicating the boundary between two constituents (**split point markers**) are not included in the **constituent alphabet**. 185, 186, 403, 415

### constituent equivalent

A **cross-lingual equivalent** (one or more possibly discontinuous aligned **words** (see **aligned word set**)) of a **target constituent**. 18, 73, 131, 139, 145, 277, 279, 280, 314, 315, 345, 346, 351, 352, 364, 365, 369, 399

### constituent form

The **constituent-inflected word** form of a **constituent** lexeme. 5, 22, 49, 52, 107, 156, 157, 159, 162, 164, 166, 168, 169, 172, 182–188, 190–192, 197, 201, 206, 208, 209, 213–216, 221, 222, 231, 251, 253, 256, 372–378, 403, 407, 411, 414, 415

### constituent inflection

The morphological transformation of a **constituent** (usually the **modifier**) during **compounding** (e.g., the addition of a **linking element** as in the German *Armut*s ‘poverty+s’ as in *Armutsbekämpfung* ‘poverty elimination’). Lieber and Štekauer (2009) call it ‘compound-specific inflection’. **Constituent inflection** is the inverse of **constituent normalization**. 15, 18, 19, 22, 31, 49, 52–54, 60, 71, 107, 158, 160, 161, 163–165, 168, 170, 171, 176, 180–183, 188–190, 193, 194, 197, 202–204, 207, 211, 214, 216, 217, 221, 223, 231, 237, 242, 255, 257, 259–263, 265, 268, 270, 271, 362, 364, 371, 373, 376, 377, 403, 404, 407, 410, 411, 415, 417

### constituent lemma

The [lemma](#) of a [constituent](#), without results of [constituent inflection](#). 4, 5, 107, 156–159, 162, 163, 166–176, 182, 184, 188, 196–202, 204–206, 208, 209, 211, 214, 216, 220, 223, 227, 230, 232, 237, 238, 242, 244, 250, 251, 256, 260, 265, 267, 269, 271, 364, 372, 373, 375–377, 403, 407, 409, 411

### constituent swapping

The phenomenon that two semantically equivalent expressions (e.g., translated [nominal compounds](#)) realize the [head](#) of the other’s expression as [modifier](#) and vice versa. For example, *draft<sub>1</sub> treaty<sub>2</sub>* translated to German as *Verfassung<sub>2</sub>entwurf<sub>1</sub>*. This phenomenon is discussed in more detail in Section 5.3.3. 26, 70, 76, 309, 310, 344

### constituent type

There are two types of [constituents](#): [modifiers](#) (usually the non-final elements of a [compound](#)) and the [head](#) (usually the final element). 25, 32, 75, 88, 162, 193, 235, 266

### content word

A [word](#) category providing semantic content, e.g., [noun](#), [verb](#), [adjective](#), [proper name](#), [adverb](#), . . . . The complement of [content words](#) are [function words](#). 2, 50, 51, 139, 168, 176, 184, 192, 212, 213, 248, 256, 281, 315, 316, 318, 348, 351, 352, 370, 398, 404, 407

### cross-lingual

[Cross-lingual](#) methods make use of properties and information derived from different (possibly aligned) [support languages](#). We use ‘[cross-lingual](#)’ when referring to a perspective across several [support languages](#), given a [target language](#). 4, 13–22, 25, 26, 36, 65, 69–73, 75–77, 79, 81–87, 91, 92, 96, 97, 101, 111, 112, 120, 132, 135, 136, 138, 140, 141, 146, 151, 153, 169–171, 225, 278–282, 285, 290, 301, 303, 304, 306, 307, 309, 310, 312–315, 317, 320, 321, 323–326, 341–349, 353, 357, 358, 361–365, 369, 390, 394, 398, 399, 402, 404, 405, 408, 411, 416

### cross-lingual equivalent

A [word](#) or [word](#) sequence that corresponds (or is *equivalent*) to the [target compound](#) (e.g., translations of a [target compound](#) in a [parallel corpus](#)). Some [equivalents](#) are phrases ([phrasal equivalents](#)) and others are [compounds](#) ([compound](#)

equivalents). 4, 14–19, 25, 66–68, 71, 72, 79, 81, 83, 84, 87, 138, 140, 150–154, 278, 310, 314, 315, 325, 329, 331, 332, 338, 344, 362, 363, 365, 370, 379, 399, 402, 403, 405, 413, 418

### cross-lingual supervision

Cross-lingual supervision is a kind of indirect supervision. An NLP method that is based on cross-lingual supervision exploits evidence about a target’s properties across languages (e.g., indicators for compound properties as occurring in parallel data). 12, 14, 15, 17, 18, 77, 81, 84, 86, 111, 131, 150, 151, 153, 282, 345, 346, 364, 367, 405, 408

## D

### dependency model

A parsing model for ternary compounds (A B C), where the dependent constituents are compared, i.e.,  $AM(A,B)$  vs.  $AM(A,C)$ . It was initially proposed by Lauer (1995b). 282–284, 287, 298, 311, 333, 343, 354, 356, 357, 383

### deterministic bottom-up parsing

A cross-lingual compound parsing method presented in Section 25.2. Starting with atomic constituents, the method iteratively merges two adjacent constituents having the smallest aligned word distance (AWD) until there is only one constituent left. 22, 344, 362, 365, 383, 411

### discovery

The task of collecting types of a certain target out-of-context, e.g., a list of non-compositional compounds. 85–88, 90–98, 100, 104, 137, 150, 362

## E

### endocentric compound

“A type of compound in which one member functions as the head and the other as its modifier, attributing a property to the head. The relation between the members of an endocentric compound can be schematized as ‘AB is (a) B’. EXAMPLE: the English compound *steamboat* as compared with *boat* is a modified, expanded version of *boat* with its range of usage restricted, so that *steamboat* will be found in basically the same semantic contexts as the noun *boat*. The compound also retains

the primary syntactic features of *boat*, since both are nouns. Hence, a *steamboat* is a particular type of *boat*, where the class of *steamboats* is a subclass of the class of *boats*. See [exocentric compound](#).” (Online Lexicon of Linguistics) . 8, 33, 37–39, 52, 98, 99, 189, 401

### Europarl Nominal Compound Database

The [Europarl Nominal Compound Database \(ENCD\)](#) is a database of English [nominal compounds](#) and their translations in up to nine European languages, as occurring in the [parallel corpus](#) EUROPARL. It has been compiled by Ziering and Van der Plas (2014) and is presented in Chapter 12. 19, 21, 79, 84, 86, 141, 147, 151, 307, 362, 379, 384, 406

### exocentric compound

“A [term](#) used to refer to a particular type of [compound](#), viz. [compounds](#) that lack a [head](#). Often these [compounds](#) refer to pejorative properties of human beings. A Dutch [compound](#) such as *wijsneus* ‘wise guy’ (lit: ‘wise-nose’) (in normal usage) does not refer to a nose that is wise. In fact, it does not even refer to a nose, but to a human being with a particular property. An alternative [term](#) used for [compounds](#) such as *wijsneus* is [bahuvrihi compound](#).” (Online Lexicon of Linguistics<sup>1</sup>) . 8, 33, 37–39, 41, 52, 70, 98, 401, 406

### extrinsic evaluation

An [extrinsic evaluation](#) is a task-based evaluation method that measures the usability of a system designed for a task  $\alpha$  on another task  $\beta$ . For example, [compound splitting](#) can be evaluated [extrinsically](#) on the task of SMT. 21, 157, 159, 162, 163, 165–169, 171, 173, 175–179, 196, 239, 241, 243, 245, 246, 248, 250, 256, 258, 267, 269, 362, 406, 409

## F

### false splitting

An erroneous behavior of a [compound splitter](#) where [compounds](#) are split into false [constituents](#). While the number of [split points](#) can be correct, these are set falsely, e.g., *Eidotter* is falsely split into *Eid | otter* ‘oath otter’ rather than into *Ei | dotter* ‘egg yolk’, or the correctly identified [constituent forms](#) are falsely

---

<sup>1</sup><http://www2.let.uu.nl/Uil-OTS/Lexicon/>

normalized. Alternative types of [compound splitting](#) errors are [undersplitting](#) and [oversplitting](#). [163](#), [208](#), [223](#), [243–245](#), [258](#), [261](#), [268](#), [269](#)

### function word

A [word](#) of the functional categories [determiner](#), [preposition](#), [conjunction](#), [pronoun](#), ...; providing no or few semantic content. The complement is a [content word](#). [29](#), [73](#), [133](#), [134](#), [184](#), [191](#), [192](#), [202](#), [213](#), [307](#), [315](#), [346](#), [348](#), [351](#), [352](#), [369](#), [370](#), [398](#), [401](#), [404](#), [417](#)

## G

### gold-constituent MOP

The [Morphological Operation Pattern](#) (MOP) which is derived from gold a [compound splits](#) (e.g., from the GermaNet [compound gold standard](#)), i.e., from a pair of true [constituent lemma](#) and related [constituent form](#). [181](#), [201](#), [202](#), [207](#), [213–216](#), [218–220](#), [257](#), [259–262](#), [396](#)

## H

### H coverage

In the task of [Recognizing Textual Entailment](#) (RTE), the ratio of lexical material from the hypothesis  $H$  being covered by the text  $T$ . [240](#), [242](#), [244](#), [248](#), [251–253](#)

### hand-crafted constituent MOP

The [Morphological Operation Pattern](#) (MOP) which describes a [constituent inflection](#) operation and is manually implemented. [182](#), [213–216](#), [219](#), [257](#)

### head

Usually the last [constituent](#) of a [compound](#) is its [head](#). More details are given in Section 3.6.1. [5](#), [6](#), [8–10](#), [24](#), [25](#), [28](#), [29](#), [32–34](#), [37](#), [38](#), [40–42](#), [47–50](#), [52–54](#), [56](#), [60–63](#), [68](#), [70](#), [71](#), [73](#), [75](#), [79](#), [95](#), [96](#), [99](#), [101](#), [102](#), [104](#), [106](#), [116](#), [122](#), [124](#), [125](#), [129](#), [134](#), [135](#), [153](#), [162](#), [168](#), [170](#), [171](#), [174–176](#), [183](#), [184](#), [189](#), [193](#), [194](#), [197](#), [202](#), [204](#), [210–214](#), [216](#), [217](#), [223](#), [226–229](#), [234–238](#), [241–243](#), [254](#), [256](#), [258](#), [266](#), [267](#), [272](#), [276](#), [284](#), [286](#), [293](#), [296](#), [301](#), [306–310](#), [346](#), [369](#), [374–378](#), [390](#), [398](#), [399](#), [404](#), [406](#), [407](#), [410](#), [411](#), [413](#), [416](#), [418](#)



## hyphenated

A lexical unit is **hyphenated** if some of its **constituents** are connected via a hyphen. 3, 4, 14, 25, 29, 31, 32, 54, 55, 59, 75, 96, 116, 122, 137, 168, 220, 232, 400, 408

## I

## identification

The task of **identifying** a certain **target in context**, e.g., **compounds** in a given sentence or the tagging of named entities. 3, 4, 14, 16–22, 26, 31, 65, 71, 76, 77, 79–92, 94, 96–99, 101–103, 106–109, 113, 115, 119–124, 130, 135, 136, 140, 141, 147–154, 172, 186, 329, 341, 362, 364–367, 370, 408

## immediate constituent

While all **atomic** parts are **constituents** of the entire **target compound**, an **immediate constituent** is directly derived from the **root node** of a **parse tree**. For example, for a LEFT-branching **3NC**, [A B]C, the **constituent** A B is the **immediate constituent** of A B C. whereas A and B are mediate **constituents**. 3, 5, 8, 42, 106, 107, 156, 184, 234–236, 276, 287, 317, 355, 403, 408, 414

## indirect supervision

Instead of using direct training data, labeled with information for the direct (underlying) task, **indirect supervision** allows for getting comparable information indirectly from task-independent data (e.g., expressive translations in a **parallel corpus**) using a *transfer function* (e.g., a **cross-linguistic** theory). This way, the task-independent information can be used indirectly as training data for the underlying task. An example of **indirect supervision** is the **cross-lingual supervision**. 12, 14, 16–18, 86, 130, 132, 161, 259, 278, 282, 345, 364, 365, 405, 408, 415

## internal structure

The **internal structure** is the result of the **structural analysis** of a **compound** and provides information *which lexemes* are *how* composed. 3, 4, 10, 71, 276, 279, 281, 282, 309, 317, 325, 339, 346, 347, 365, 402, 408

## intrinsic evaluation

An **intrinsic evaluation** (in contrast to an **extrinsic evaluation**) is an evaluation method that measures the performance of a system designed for a task  $\alpha$  directly

on this task, using common measurements like accuracy, precision or recall. 157, 159, 162, 166–178, 195, 198, 200, 208, 231, 245, 256–259, 267, 268, 271, 272, 409

## L

### leaf node

A **leaf node** is the final node of a **parse tree** which has no outgoing branches (i.e., the opposite of a **root node**). 195, 196, 328, 329, 335, 336, 355, 374, 409, 414

### LEFT class baseline

This is a simple major class baseline for the **parsing** of **ternary compounds**, i.e., for a binary classification (LEFT vs. RIGHT). As observed in many previous work, the major class is LEFT, with a percentage of about 64% (Resnik, 1993), 66% (Lauer, 1994, Lauer and Dras, 1994) or 67% (Lauer, 1995a). 186, 236, 266, 289, 290, 294–296, 301, 313, 323, 324, 339, 341, 344

### lemma

A concrete and representative embodiment of a **lexeme**. It is the basic and uninflected **word form** as listed in a dictionary. 5, 19, 49, 52, 53, 84, 87, 94, 97, 107, 113, 146, 156, 159, 160, 165, 169, 170, 181, 183–194, 197–199, 201, 202, 204, 206, 207, 210–219, 223, 227–229, 232, 233, 238, 242, 247, 248, 250, 251, 253, 256, 257, 263, 267, 269, 270, 372–378, 390, 392, 396, 399, 404, 411, 419

### lemma sequence format

The **lemma sequence format** (LSF) is the representation of a **compound split** as a sequence of **constituent lemmas**, separated by space, e.g., the **split** of *Hühnersuppe* ‘chicken soup’ is represented in the LSF as *Huhn Suppe*. 196, 242, 256, 385, 409

### lexeme

“The **lexeme** is defined as a set of syntactic and semantic features shared by one or several morpho-syntactic elements. Roughly speaking, it contains the kind of information one expect to find in a standard dictionary entry” (Wehrli, 1985) . 2, 4–7, 17, 24, 29, 30, 34, 40, 48, 53, 56–60, 70, 76, 130, 164, 184, 191–194, 199, 212–216, 240, 242–245, 247, 256, 268, 270, 273, 401, 402, 408, 409, 414, 415, 417, 418

### linguistic criterion

A [linguistic criterion](#) is a test for validating a certain property of a linguistic expression, e.g., a [criterion](#) for [compoundhood](#). If the conditions described in the [criterion](#) are met, we have evidence for the underlying property (e.g., [compoundhood](#)). [4](#), [11](#), [15](#), [17](#), [19–21](#), [25](#), [56](#), [58–64](#), [75–77](#), [80–84](#), [115–118](#), [120](#), [123](#), [124](#), [127–132](#), [149–154](#), [362](#), [363](#), [366](#), [384](#), [410](#)

### linking element

The most frequent type of [constituent inflection](#) is the suffixation of the [modifier](#). The morpheme which is added between [modifier](#) and [head](#) is called [linking element](#) (e.g., the *s* in German, the so-called *Fugen-s*, see Section [3.9.2](#)). [11](#), [15](#), [19](#), [31](#), [47](#), [49](#), [52](#), [54](#), [58](#), [60](#), [61](#), [143](#), [146](#), [157](#), [158](#), [164](#), [165](#), [169–173](#), [182](#), [203](#), [206](#), [207](#), [259](#), [364](#), [378](#), [403](#), [410](#), [416](#), [417](#)

## M

### modifier

Usually the non-final [constituent](#) of a [compound](#) is (one of) its [modifiers](#). More details are given in Section [3.6.3](#). [5](#), [10](#), [24](#), [25](#), [28](#), [29](#), [31–34](#), [37](#), [39](#), [40](#), [42](#), [47–54](#), [57](#), [58](#), [60–64](#), [68–71](#), [73](#), [75](#), [88](#), [99](#), [101](#), [102](#), [104](#), [106](#), [116](#), [123–125](#), [128–131](#), [134](#), [141](#), [143](#), [146](#), [151](#), [159](#), [162](#), [165](#), [170–172](#), [175](#), [176](#), [183](#), [189](#), [193](#), [194](#), [201–204](#), [206](#), [210](#), [212–216](#), [218](#), [220](#), [221](#), [226–229](#), [234–238](#), [241–243](#), [256](#), [258](#), [262](#), [263](#), [266](#), [267](#), [269](#), [270](#), [272](#), [276](#), [284](#), [286](#), [294](#), [297](#), [298](#), [301](#), [302](#), [306](#), [307](#), [309](#), [310](#), [378](#), [394](#), [399](#), [403](#), [404](#), [406](#), [410](#), [413](#)

### MOP application

The application of an [Morphological Operation Pattern](#) (MOP) on a string  $\Sigma$  resulting in a string  $\Omega$ . [182](#), [183](#), [187](#), [188](#), [192](#), [194](#), [256](#), [261](#), [270](#), [271](#), [377](#), [390](#), [391](#)

### Morphological Operation Pattern

The [Morphological Operation Pattern](#) (MOP) is an ordered list of context-free substring replacements, allowing for modelling many morphological transformations. More details about [MOPs](#) are given in Chapter [17](#). [21](#), [160](#), [163](#), [180](#), [184](#), [256](#), [362](#), [385](#), [407](#), [410](#), [412](#), [419](#)

## multilingual

**Multilingual** methods make use of universal properties of any natural languages and therefore can be seen as kind of language-independent (within a certain scope). **Multilingual** methods can be applied (or easily adapted) to multiple **target languages**. 12, 19, 21, 75, 77, 79, 81, 86, 89, 93, 94, 97, 107, 138, 160, 161, 163, 165, 169, 170, 172, 178, 184, 247, 255–258, 262, 269, 304, 362, 364, 367, 411

## N

### nominal compound

The main subject of this thesis: a **compound** with a nominal **head** (e.g., *database* or *hot dog*). 3, 6, 8, 17–20, 28, 29, 37, 41, 46–48, 50, 51, 57, 68, 75, 79, 80, 82, 84–86, 89, 90, 94–97, 99, 106–108, 113, 117–119, 122, 124, 133, 135, 137, 139–141, 144, 150, 152–154, 200, 205, 275, 306, 308, 314, 366, 370, 391, 404, 406, 412

### non-deterministic full tree accumulation parsing

A **cross-lingual compound parsing** method presented in Section 25.3.2. In contrast to the **deterministic bottom-up parsing**, this method enumerates all possible **binary parse trees** and validates them according to a tree annotation principle based on **AWD**. 22, 326, 344, 362, 386

### non-split option

A **splitting** analysis without any **split point**, i.e., the **target** expression is considered as **atomic**. 167, 185, 186, 189, 190, 194, 227, 256

### normalization

The process of mapping a **constituent form** to the underlying **constituent lemma**. While **lemmatization** is designed for regular **word inflection**, **normalization** addresses **constituent inflection** and thus also includes non-paradigmatic transformations. 5, 156, 157, 159, 162–164, 166, 167, 170, 172, 183–187, 190–192, 194, 196, 197, 204, 205, 209, 210, 213–216, 218, 220, 221, 223, 227, 231, 233–235, 237, 253, 256, 257, 262–264, 267–270, 364, 391, 403, 404, 407, 411

### noun compound

A **compound** composed of only nouns (e.g., *telephone cable* but not *hot dog*). 9, 17, 18, 28, 29, 37, 42, 48, 53, 59, 82, 85, 90, 95, 99–105, 108, 109, 111, 122, 146, 175, 276, 292, 294, 301, 303, 343, 369, 413

### **noun phrase**

A lexical phrase having a noun as head. Optional attributes include adjectives or genitives (usually prenominal), and prepositional phrases, genitive NPs or relative clauses (usually postnominal). For example [*Peter's brown dog that I saw yesterday*], where the head *dog* has preceding and succeeding attributes. All kinds of [nominal compounds](#) are also considered as [noun phrases](#). A [noun phrase](#) without any trailing attributes is called [base noun phrase](#). [43](#), [111](#), [281](#), [300](#), [386](#), [399](#), [412–414](#)

### **null-MOP**

The [Morphological Operation Pattern](#) (MOP) which does not contain any substring replacements  $\mu$ , i.e., it represents the morphological null-operation, that does not alter the string. [191](#), [193](#), [194](#), [213](#), [214](#), [216](#), [217](#), [219](#), [221](#), [257](#)

## **O**

### **open compound**

A multi-word [compound](#), i.e., a [compound](#) spelled with several whitespace-separated [words](#) (e.g., *natural language processing*). [4](#), [5](#), [14](#), [18](#), [24](#), [25](#), [27](#), [29](#), [31](#), [32](#), [59](#), [64](#), [66](#), [75](#), [80](#), [85](#), [90](#), [95](#), [96](#), [107](#), [108](#), [112](#), [116](#), [134](#), [137](#), [156](#), [169](#), [175](#), [234](#), [242](#), [245](#), [251](#), [281](#), [294](#), [412](#)

### **open compounding**

A language's property of creating [open compounds](#). [137](#), [157](#), [367](#), [412](#)

### **open compounding language**

A language with [open compounding](#), i.e., in which [compounds](#) are usually [open](#), such as *English*. [31](#), [32](#), [46](#), [59](#), [127](#), [131](#), [134](#), [151](#), [156](#), [165](#), [281](#)

### **oversplitting**

An erroneous behavior [compound splitter](#) where [targets](#) are [split](#) into too many [constituents](#) (e.g., a recursive [splitting](#) process ends too late). The contrary behavior is called [undersplitting](#). [160](#), [163](#), [176](#), [196](#), [208](#), [243–245](#), [253](#), [258](#), [261](#), [268](#), [269](#), [272](#), [407](#), [417](#)

## **P**

### parallel corpus

A [parallel corpus](#) is a corpus that contains the same semantic content in various languages such that sentences (or even [words](#)) can be cross-lingually aligned. [12–15](#), [17](#), [19–21](#), [26](#), [34](#), [65–68](#), [71](#), [72](#), [77](#), [79](#), [80](#), [83](#), [84](#), [86](#), [87](#), [91–94](#), [100](#), [101](#), [111–114](#), [116](#), [132](#), [136](#), [138](#), [140](#), [141](#), [147](#), [150](#), [151](#), [160](#), [169–171](#), [179](#), [277–279](#), [281](#), [282](#), [303](#), [304](#), [324](#), [335](#), [342](#), [357](#), [358](#), [362](#), [363](#), [365](#), [394](#), [405](#), [406](#), [408](#), [413](#), [416](#), [418](#)

### parse tree

A (usually [binary](#)) tree structure representation of a [parse](#). [5](#), [18](#), [22](#), [277](#), [280](#), [282](#), [297](#), [300](#), [301](#), [303](#), [304](#), [317–320](#), [323](#), [326–329](#), [331–341](#), [344](#), [345](#), [348](#), [349](#), [354–356](#), [358](#), [392](#), [397](#), [400](#), [402](#), [408](#), [409](#), [411](#), [414](#)

### phrasal compound

A [compound](#) with a phrasal [modifier](#) (e.g., the *[do-it-yourself strategy](#)*) (Meibauer, 2003). [25](#), [34](#), [39](#), [48](#), [59](#), [118](#), [201](#), [269](#)

### phrasal equivalent

A [cross-lingual equivalent](#) which is a phrase (i.e., an [aligned phrase](#)). [68](#), [71–74](#), [153](#), [277](#), [279](#), [280](#), [306](#), [335](#), [338](#), [344](#), [345](#), [347](#), [348](#), [351](#), [352](#), [354](#), [356](#), [365](#), [369](#), [394](#), [398](#), [405](#), [416](#)

### PoS pattern

A predefined sequence of [PoS](#) tags modeling a certain kind of linguistic expression, e.g., [noun compounds](#) or [base noun phrases](#). [22](#), [87–89](#), [92–95](#), [109](#), [132](#), [137](#), [141–143](#), [145](#), [148](#), [149](#), [151](#), [153](#), [154](#), [306](#), [321](#), [338](#), [369](#), [370](#), [395](#), [414](#), [417](#)

### prepositional phrase

A lexical phrase having a preposition as [head](#) and usually a [noun phrase](#) as complement. For example [*[for](#) [Peter's brown dog that I saw yesterday]*]. [10](#), [43](#), [387](#)

## R

### root node

A [root node](#) is the initial node of a [parse tree](#) which has no incoming branches (i.e., the opposite of a [leaf node](#)). [128](#), [327](#), [331](#), [335](#), [344](#), [345](#), [356](#), [408](#), [409](#), [414](#)

## S

**semantic association**

The association strength of two [constituents](#) with respect to semantics (e.g., [constituents](#) forming a common [lexeme](#)). 11, 18, 221, 276–278, 281–283, 285, 306, 315–317, 322, 343, 344, 346, 348, 349, 352, 353, 355, 357, 364, 365

**semantic indeterminacy**

The property of semantic equivalence between structural analyses of a linguistic expression (e.g., sentence, [noun phrase](#) or *k*-[noun Compound](#) (kNC)). For example, the 3NC *college football player* is [semantically indeterminate](#) - it can be considered as being “both LEFT- and RIGHT-branching, i.e. a dependency should exist between all [word pairs](#)” (Vadas, 2009): a *player* of *college football*  $\equiv$  a *football player* attending a *college*. More details are discussed in Section 3.8.3. 25, 26, 36, 43–45, 65, 72, 73, 75, 77, 235, 279, 280, 289, 298, 312, 318, 321, 329, 332, 333, 336, 338–343, 345, 348, 349, 397, 414

**semantic lexicon bootstrapping**

A semi-supervised approach of learning semantic lexicons (e.g., lists of [terms](#) of a certain semantic class such as SUBSTANCE) automatically from a given seed lexicon and usually contextual properties (e.g., represented as *N*grams, [PoS patterns](#) or lexico-syntactic patterns). 93, 94, 314, 349–351

**semantic relation**

The relation that holds between two ([immediate](#)) [constituents](#) of a [compound](#), e.g., the `made_out_of` relation for *gold ring* (i.e., a ring made out of gold). 5, 8, 13, 14, 18, 25, 26, 42, 47, 62, 65, 73–75, 81, 95, 100, 102, 170, 276, 293, 302, 348, 366, 367, 369, 388

**split point**

The [split point](#) of a [closed compound](#) is the position in the [word](#) at which two adjacent [constituent forms](#) are concatenated. During [compound splitting](#), the [split point](#) is highlighted by the pipe symbol (in the [split point format](#) (SPF)), as in *Hühner | suppe* ‘chicken soup’. 5, 8, 19, 31, 71, 84, 97, 157, 159, 162, 163, 166, 168, 171, 172, 176, 178, 184–186, 189, 196, 197, 200, 205, 206, 208, 209, 211, 213, 214, 216–223, 231, 235, 256, 257, 262–264, 267, 268, 270, 272, 364, 370, 372, 374, 378, 407, 411, 414, 415, 417

### split point format

The **split point format** (SPF) is a representation of a **compound split** as a sequence of **constituent forms** separated by the pipe symbol, e.g., *Hühner|suppe* ‘chicken | soup’. 22, 186, 191, 196, 197, 256, 372, 388, 415

### split point marker

A **split point marker** is a special character (not part of the **constituent alphabet**) that marks the boundary between two constituents, i.e., the **split point**. The (almost exclusive) representative of a split point marker is the **hyphen** as in *TV-Programm* ‘TV program’. 32, 185, 186, 220, 232, 253, 269, 290, 403, 415

### split tree

A **split tree** is a tree structure of a **closed compound**, e.g., as output of a recursive (binary) **compound splitter**. 167, 184, 190, 195, 196, 199–201, 219, 233, 256, 271, 272, 374–376, 391, 392, 415

### structural analysis

The **structural analysis** of **compounds** includes the determination of the composed **lexemes** (in the case of **closed compounds**, i.e., **compound splitting**) and the way how these **constituents** are combined (in the case of **compounds** comprising three or more **constituents**, i.e., **compound parsing**). 3–5, 17, 20, 21, 269, 315, 361, 362, 366, 367, 388, 401, 408, 415

### subtree

A subtree *st* of a full tree *ft* is a tree consisting of a node in *ft* and all of its descendants. This means, the full tree *ft* is the largest **subtree** *st* of *ft*. 327, 333, 335–337, 342, 345, 354, 415

### supervision based on morphological regularities

A kind of **indirect supervision** that exploits morphological regularities. In this thesis, we use the regularity of **constituent inflection** sharing many operations with regular **word inflection** (e.g., the addition of **linking element**). 12, 14, 364

### support language

The counterpart to a **target language** in a **cross-lingual** task. While a **target language** is the language of a **target compound**, the **support languages** are usually



aligned to the **target** in a **parallel corpus**, providing **cross-lingual** evidence (e.g., expressive **word** positions of a **phrasal equivalent** for a **target compound**) . 83, 85, 86, 136, 138–140, 146, 152, 169, 280, 281, 307, 310, 311, 315, 317, 318, 322, 325–329, 332, 335–338, 341, 342, 344, 345, 348, 349, 357–359, 369, 404, 416

**synthetic compound**

A **compound** with a deverbal **head** (e.g., *truck driver*). 9, 10, 42, 47, 52, 153

**T**

**target compound**

The **target compound** is the target of an underlying **compound** process, e.g., the subject of **compound parsing** or **compound splitting**. 4, 5, 7, 14, 16, 25, 41, 80, 81, 85, 87, 115, 131, 151, 153, 161, 169–172, 184, 191, 192, 199–204, 211, 213, 219, 226–235, 244, 256, 257, 261, 265, 269, 279, 281, 306, 309, 310, 314, 315, 317, 318, 322, 325, 328, 329, 335, 336, 340, 343, 344, 347, 351, 358, 363–365, 372–376, 378, 398, 405, 408, 411, 412, 416

**target constituent**

The **constituents** of a **target compound**. 18, 281, 285, 306, 311, 315–320, 322, 323, 325–327, 335, 344, 346–348, 351, 353, 354, 358, 365, 399, 403

**target language**

A **target language** is the language of the **target compound** a **cross-lingual** method is applied to, e.g., for **parsing English compounds** with the usage of various **support languages**. 12, 18, 19, 75, 83, 85, 86, 96, 100, 101, 113, 117, 136, 138, 140, 165, 167–171, 173–176, 197, 198, 255, 263, 281, 297, 315, 358, 363, 367, 404, 411, 416

**term**

A linguistic expression covering single **words**, phrases or **MWEs**. 43, 47, 58, 92–95, 225, 226, 242, 245, 312, 402, 406, 414

**ternary compound**

A **compound** with three **constituents** (e.g., *human<sub>1</sub> rights<sub>2</sub> abuse<sub>3</sub>*). 29, 44, 89, 137, 269, 275, 283, 309, 312, 343, 346, 388, 398, 405, 409

## token

A concrete instantiation of an object (e.g., a [word](#) or [compound](#)) in context. The [token](#) frequency of a [word](#) in a corpus is the count of all of its occurrences. [7](#), [15](#), [29](#), [30](#), [72](#), [86](#), [119](#), [120](#), [124](#), [147](#), [172](#), [198](#), [199](#), [202](#), [242–245](#), [247](#), [248](#), [251–253](#), [268](#), [269](#), [278](#), [279](#), [282](#), [306](#), [312](#), [316](#), [318](#), [321–324](#), [329](#), [331](#), [336](#), [338](#), [341](#), [343](#), [346](#), [347](#), [357](#), [417](#)

## type

The context-independent representation of an object (e.g., a [word](#) or [compound](#) in a dictionary). [6](#), [7](#), [29](#), [30](#), [131](#), [199](#), [277](#), [279](#), [282](#), [318](#), [321–325](#), [329](#), [336](#), [338](#), [341](#), [347](#), [348](#), [354](#), [357](#), [365](#), [397](#), [405](#)

## U

### undersplitting

An erroneous behavior of a [compound splitter](#) where [compounds](#) are [split](#) into too few parts or do not get any [split point](#) at all (e.g., a recursive [splitting](#) process ends too early). The contrary behavior is called [oversplitting](#). [133](#), [160](#), [163](#), [167](#), [168](#), [175](#), [196](#), [203](#), [208](#), [216](#), [218](#), [220–223](#), [243](#), [244](#), [253](#), [257](#), [258](#), [261–263](#), [268](#), [269](#), [272](#), [396](#), [407](#), [413](#)

### universal surface pattern

A [universal surface pattern](#) (USP) is a universally valid and generalized [PoS pattern](#), where sequences of [function words](#) and [compound splits](#) get a special representation. More details about [USPs](#) are given in [Appendix A](#). [22](#), [132](#), [144](#), [306](#), [389](#), [417](#)

### unparadigmatic

A [constituent inflection](#) operation (e.g., the addition of a [linking element](#)) applied to a [lexeme](#)  $\gamma$  which does not correspond to any [word inflection](#) operation of  $\gamma$  is called ‘unparadigmatic’ (Neef, 2009). For example, the [linking element](#)  $\oplus_s$  in the German *Armuts* ‘poverty+s’ as in *Armutsbekämpfung* ‘poverty elimination’. [49](#)

## V

**verb compound**

A [compound](#) composed of only verbs (e.g., *to freeze-dry* but not *to fingerprint*).  
29

**verbal compound**

A [compound](#) with a verbal [head](#) (e.g., *to fingerprint*) . 29, 37, 46–48, 51, 79, 95, 97, 366

**W**

**word**

A [word](#) is an isolated unit, occurring typically separated by whitespace or punctuation characters (e.g., a [closed compound](#)). 2, 3, 6–11, 13–15, 18, 24, 27, 29–32, 40, 42, 44, 50, 52–64, 66, 68, 71, 72, 76, 79, 83, 88–94, 97, 101, 104, 105, 108, 111, 115, 116, 118, 127, 130–133, 135, 137–140, 143–146, 151–153, 159, 160, 166, 169, 171, 173, 180, 184, 189, 191, 192, 199, 203, 205, 206, 208, 219, 222, 225, 232, 240, 241, 243, 250, 259–261, 268, 270, 272, 276, 282, 285–287, 290, 291, 293, 294, 296–304, 308, 315, 316, 319, 325, 328, 348, 350, 351, 353, 354, 357, 358, 363, 370, 373, 376–380, 400, 401, 403–405, 407, 412–414, 416–418

**word alignment**

The (commonly automatic) process (or result) of mapping [words](#) from one language to their [equivalents](#) in another language, in a [parallel corpus](#). 66, 68, 113, 114, 132, 134, 170, 175, 310, 311, 313, 315, 323, 324, 344, 347, 418

**word alignment error**

False mappings in the [word alignment](#) (e.g., missing or spurious links). 133, 134, 347, 357

**word distance**

The distance between two units  $\phi$  and  $\xi$  in terms of [words](#), i.e., the number of [words](#) between  $\phi$  and  $\xi$ , plus 1. 277–279, 345, 346, 364, 365

**word form**

A [word form](#) is a possibly [word-inflected](#) embodiment of a [lexeme](#) in context. 19, 84, 172, 181, 183, 190, 192, 194, 227, 256, 270, 409, 418, 419

**word inflection**

The regular morphological transformation marking morpho-syntactic functions such as case, gender or number (e.g., *Kartoffeln* ‘potatoes’ as the pluralized form of *Kartoffel* ‘potato’). Lieber and Štekauer (2009) call it ‘non-compound specific inflection’. 11, 15, 19, 47, 54, 60, 61, 99, 116, 160, 161, 165, 169, 181, 183, 184, 188, 190, 191, 193, 194, 197, 198, 202–204, 207, 210–212, 214, 215, 217, 231, 237, 242, 253, 255–257, 259–263, 265, 268, 270, 271, 362, 364, 373, 376, 377, 411, 416–419

**word MOP**

The **Morphological Operation Pattern** (MOP) which is learned automatically from **word inflection**, i.e., from a pair of **lemma** and **word form**. 181, 187–194, 199, 201, 207, 209, 211–221, 223, 231, 232, 237, 250, 253, 257, 259, 260, 262, 263, 265, 268, 270, 396

# Bibliography

- Adda-decker, M., Adda, G., and Lamel, L. (2000). Investigating text normalization and pronunciation variants for German broadcast transcription. In *In ICSLP 2000*, pages 266–269.
- Alfonseca, E., Bilac, S., and Pharies, S. (2008). *German Decompounding in a Difficult Corpus*, pages 128–139. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Algeo, J. and Algeo, A. S. (1993). *Fifty Years Among the New Words: A Dictionary of Neologisms 1941-1991*. Centennial series of the American Dialect Society. Cambridge University Press.
- Arts, T. (2014). The Making of a Large English-Arabic/Arabic-English Dictionary: the Oxford Arabic Dictionary. In Abel, A., Vettori, C., and Ralli, N., editors, *Proceedings of the 16th EURALEX International Congress*, pages 109–124, Bolzano, Italy. EURAC research.
- Augst, G. (1975). *Über das Fugenmorphem bei Zusammensetzungen*. Untersuchungen zum Morpheminventar der deutschen Gegenwartssprache. Narr, Tübingen.
- Augustinus, L. and Dirix, P. (2013). The IPP effect in Afrikaans: A Corpus Analysis. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 213–225.
- Baayen, H. R., Piepenbrock, R., and Gulikers, L. (1995). *The CELEX Lexical Database*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania.
- Baldwin, T. and Kim, S. N. (2010). Multiword Expressions. In Indurkha, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL. ISBN 978-1420085921.
- Baldwin, T. and Tanaka, T. (2004). Translation by machine of complex nominals: Getting it right. In *Proceedings of the ACL04 Workshop on Multiword Expressions: Integrating Processing*.
- Bally, C. (1944). *Linguistique générale et linguistique française*. P.U.F.
- Barker, K. (1998). A Trainable Bracketer for Noun Modifiers. In *Canadian Conference on AI*, Lecture Notes in Computer Science.

## BIBLIOGRAPHY

- Baroni, M., Matiassek, J., and Trost, H. (2002). Predicting the Components of German Nominal Compounds. In *ECAI*, pages 470–474. IOS Press.
- Barrière, C. and Ménard, P. A. (2014). Multiword Noun Compound Bracketing using Wikipedia. In *ComAComA 2014*.
- Barz, I. (2005). Die Wortbildung. In *Duden: Die Grammatik*, page 641–772. Dudenverlag, Mannheim.
- Batra, A. and Paul, S. (2015). A Hybrid Approach for Bracketing Noun Sequence. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 276–284, Trivandrum, India. NLP Association of India.
- Batra, A., Paul, S., and Kulkarni, A. (2014). *Constituency Parsing of Complex Noun Sequences in Hindi*, pages 285–296. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Bauer, L. (1983). *English Word-Formation*. Cambridge University Press, Cambridge.
- Bauer, L. (1998a). Is there a class of neoclassical compounds, and if so is it productive? *Linguistics*, 3(36):403–22.
- Bauer, L. (1998b). When is a sequence of noun + noun a compound in English? *English Language and Linguistics*, 2:65–86.
- Bauer, L. (2001). Compounding. In *Language Typology and Language Universals*. Mouton de Gruyter.
- Bauer, L. (2003). *Introducing Linguistic Morphology*. Introducing Linguistic Morphology. Edinburgh University Press.
- Bauer, L. (2006). Compound.
- Behaghel, O. (1909). Beziehungen zwischen Umfang und Reihenfolge von Satzgliedern. *Indogermanische Forschungen*.
- Bejcek, E. and Stranák, P. (2010). Annotation of Multiword Expressions in the Prague Dependency Treebank. *Language Resources and Evaluation*, 44(1-2):7–21.
- Berg-Kirkpatrick, T. and Klein, D. (2010). Phylogenetic Grammar Induction. In *ACL 2010*.
- Bergsma, S., Pitler, E., and Lin, D. (2010). Creating Robust Supervised Classifiers via Web-scale N-gram Data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL ’10, pages 865–874, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bergsma, S., Yarowsky, D., and Church, K. (2011). Using Large Monolingual and Bilingual Corpora to Improve Coordination Disambiguation. In *ACL-HLT 2011*.
- Bies, A., Mott, J., Warner, C., and Kulick, S. (2012). English Web Treebank. Technical report, LDC2012T13, Linguistic Data Consortium, Philadelphia, PA.

## BIBLIOGRAPHY

- Bikel, D. and Zitouni, I. (2012). *Multilingual Natural Language Processing Applications: From Theory to Practice*. IBM Press, 1st edition.
- Bisetto, A. and Scalise, S. (2005). The classification of compounds. *Lingue e linguaggio*, 4 (2), pages 319–32.
- Bloomfield, L. (1933). *Language*. University of Chicago Press.
- Booij, G. (1992). Compounding in Dutch. *Rivista di Linguistica*, 1(4):37–60.
- Booij, G. (1995). *The Phonology of Dutch*. Oxford linguistics. Clarendon Press.
- Booij, G. (2005). *The Grammar of Words: An Introduction to Linguistic Morphology*. Oxford linguistics. Oxford University Press.
- Botha, R. P. (1981). *A base rule theory of Afrikaans synthetic compounding*. Foris.
- Bouillon, P., Boesefeldt, K., and Russell, G. (1992). Compound Nouns in a Unification-Based MT System. In *ANLP 1992*, pages 209–215, Trento.
- Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Brants, T. and Franz, A. (2006). *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, PA. Philadelphia, PA.
- Bretschneider, C. and Zillner, S. (2015). Semantic Splitting of German Medical Compounds. In *Text, Speech, and Dialogue*. Springer International Publishing.
- Brown, R. D. (2002). Corpus-Driven Splitting of Compound Words. In *In Proceedings of the Ninth international Conference on Theoretical and Methodological Issues in Machine Translation (TMI-2002)*.
- Buckley, C. and Voorhees, E. M. (2000). Evaluating Evaluation Measure Stability. In *SIGIR 2000*.
- Burkett, D. and Klein, D. (2008). Two Languages are Better than One (for Syntactic Parsing). In *EMNLP 2008*.
- Burnard, L. (2000). User Reference Guide for the British National Corpus. Technical report, Oxford University Computing Services.
- Cap, F. (2014). *Morphological processing of compounds for statistical machine translation*. PhD thesis, Uni Stuttgart.
- Cap, F., Fraser, A., Weller, M., and Cahill, A. (2014). How to Produce Unseen Teddy Bears: Improved Morphological Processing of Compounds in SMT. In *EACL'14: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 579–587, Gothenburg, Sweden.

## BIBLIOGRAPHY

- Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.
- Caseli, H. d. M., Ramisch, C., das Gracas Volpe Nune, M., and Villavicencio, A. (2010). Alignment-based extraction of multiword expressions. *Language Resources and Evaluation*, 44(1-2):59–77.
- Caseli, H. d. M., Villavicencio, A., Machado, A., and Finatto, M. J. (2009). Statistically-driven Alignment-based Multiword Expression Identification for Technical Domains. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, MWE '09, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Celli, F. and Nissim, M. (2009). Automatic Identification of Semantic Relations in Italian Complex Nominals. In *Proceedings of the Eight International Conference on Computational Semantics*.
- Chinchor, N. A. (1998). Overview of MUC-7/MET-2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.
- Chomsky, N. and Halle, M. (1968). *The Sound Pattern of English*. Studies in language. Harper & Row.
- Church, K. and Patil, R. (1982). Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table. *Computational Linguistics*.
- Clark, E. V. (1981). Lexical innovations: How children learn to create new words. In Deutsch, W., editor, *The Child's Construction of Language*, pages 299–328. Academic Press, New York.
- Clouet, E. and Daille, B. (2014). Splitting of Compound Terms in non-Prototypical Compounding Languages. In *Workshop on Computational Approaches to Compound Analysis, ComACoMA, COLING 2014*, Dublin, Ireland.
- Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20.
- Combrink, J. (1990). *Afrikaanse morfologie: capita exemplaria*. Academica.
- Constant, M., Eryiğit, G., Monti, J., van der Plas, L., Ramisch, C., Rosner, M., and Todirascu, A. (2017). Multiword Expression Processing: A Survey. *Computational Linguistics*.
- Cook, P., Fazly, A., and Stevenson, S. (2008). The VNC-Tokens Dataset. In *Proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, Marrakech, Morocco.
- Corley, S., Corley, M., Keller, F., Crocker, M. W., and Trewin, S. (2001). Finding Syn-



## BIBLIOGRAPHY

- tactic Structure in Unparsed Corpora The Gsearch Corpus Query System. *Computers and the Humanities*, 35(2):81–94.
- Dagan, I., Dolan, B., Magnini, B., and Roth, D. (2009). Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(Special Issue 04).
- Dagan, I. and Glickman, O. (2004). Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability. In *Learning Methods for Text Understanding and Mining*.
- Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 177–190, Berlin, Heidelberg. Springer-Verlag.
- Daiber, J., Quiroz, L., Wechsler, R., and Frank, S. (2015). Splitting Compounds by Semantic Analogy. *CoRR*.
- Daumé III, H. (2004). Notes on CG and LM-BFGS Optimization of Logistic Regression. Paper available at <http://pub.hal3.name#daume04cg-bfgs>, implementation available at <http://hal3.name/megam/>.
- de Haas, W. and Trommelen, M. (1993). *Morfologisch handboek van het Nederlands: een overzicht van de woordvorming*. SDU Uitgeverij.
- Di Sciullo, A.-M. and Williams, E. (1987). *On the Definition of Word*. Linguistic inquiry monographs. MIT Press.
- Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*, Lisbon, Portugal. European Language Resources Association (ELRA).
- Don, J. (2009). I-E-Germanic: Dutch. In Lieber, R. and Štekauer, P., editors, *The Oxford Handbook of Compounding*, chapter 19. Oxford University Press, Oxford.
- Donalies, E. (2004). *Grammatik des Deutschen im europäischen Vergleich: Kombinatorische Begriffsbildung: Teil I : Substantivkomposition*.
- Donalies, E. (2005). *Die Wortbildung des Deutschen: ein Überblick*. Studien zur deutschen Sprache. Narr.
- Downing, P. (1977). On the Creation and Use of English Compound Nouns. *Language*, 53(4):810–842.

## BIBLIOGRAPHY

- Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. A Wiley Interscience Publication. Wiley.
- Durgar El-Kahlout, I. and Yvon, F. (2010). The pay-offs of preprocessing for German-English Statistical Machine Translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 251–258.
- Dyer, C. (2009). Using a Maximum Entropy Model to Build Segmentation Lattices for MT. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL '09*, pages 406–414.
- Eiselen, R. and Puttkammer, M. J. (2014). Developing Text Resources for Ten South African Languages. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 3698–3703.
- Escartín, C. P. (2014). Chasing the Perfect Splitter: A Comparison of Different Compound Splitting Tools. In *LREC 2014*.
- Evert, S. and Krenn, B. (2005). Using Small Random Samples for the Manual Evaluation of Statistical Association Measures. *Comput. Speech Lang.*, 19(4):450–466.
- Farahmand, M., Smith, A., and Nivre, J. (2015). A Multiword Expression Data Set: Annotating Non-Compositionality and Conventionalization for English Noun Compounds. In *Proceedings of the 11th Workshop on Multiword Expressions, MWE@NAACL-HLT 2015, June 4, 2015, Denver, Colorado, USA*, pages 29–33.
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press.
- Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32.
- Fossum, V. and Knight, K. (2008). Using bilingual Chinese-English word alignments to resolve PPattachment ambiguity in English. In *AMTA Student Workshop 2008*.
- Fritzing, F. and Fraser, A. (2010). How to Avoid Burning Ducks: Combining Linguistic Analysis and Corpus Statistics for German Compound Processing. In *Proceedings of the ACL 2010 Joint 5th Workshop on Statistical Machine Translation and Metrics MATR*, pages 224–234.
- Frunza, O. and Inkpen, D. (2009). Identification and disambiguation of cognates, false friends, and partial cognates using machine learning techniques. *International Journal of Linguistics*, 1(1).
- Fuhrhop, N. (1998). *Grenzfälle morphologischer Einheiten*. Studien Zur Deutschen Grammatik. Stauffenburg.
- Gao, Q. and Vogel, S. (2008). Parallel Implementations of Word Alignment Tool. In *Soft-*

## BIBLIOGRAPHY

- ware *Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57.
- Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Giegerich, H. J. (2004). Compound or phrase? English noun-plus-noun constructions and the stress criterion. *English Language and Linguistics*, 8(1):1–24.
- Girju, R. (2007). Improving the Interpretation of Noun Phrases with Crosslinguistic Information. In *ACL 2007*.
- Girju, R., Moldovan, D. I., Tatu, M., and Antohe, D. (2005). On the Semantics of Noun Compounds. 19(4):479–496.
- Graves, W. W., Binder, J. R., and Seidenberg, M. S. (2013). Noun–noun combination: Meaningfulness ratings and lexical statistics for 2,160 word pairs. *Behavior Research Methods*, 45(2):463–469.
- Grimshaw, J. B. (1990). *Argument Structure*. Linguistic inquiry monographs. MIT Press.
- Grishman, R. and Sundheim, B. (1995). Design of the MUC-6 Evaluation. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, pages 1–11, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Guevara, E., Scalise, S., Bisetto, A., and Melloni, C. (2006). MORBO/COMP: a Multilingual Database of Compound Words. In *LREC 2006*.
- Günther, H. (1981). N+N: Untersuchungen zur Produktivität eines deutschen Wortbildungstyps. In Lipka, L. and Günther, H., editors, *Wortbildung*. Foris, Dordrecht.
- Haapalainen, M. and Majorin, A. (1995). GERTWOL und Morphologische Disambiguierung für das Deutsche. Technical report.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18.
- Hamp, B. and Feldweg, H. (1997). GermaNet - a Lexical-Semantic Net for German. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.
- Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.
- Haspelmath, M. (2002). *Understanding Morphology*. Understanding Morphology. Arnold.

## BIBLIOGRAPHY

- Hellmann, S., Stadler, C., Lehmann, J., and Auer, S. (2009). DBpedia Live Extraction. In *Proc. of 8th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, volume 5871 of *Lecture Notes in Computer Science*, pages 1209–1223.
- Hendrickx, I., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Szpakowicz, S., and Veale, T. (2013). Semeval-2013 task 4: Free paraphrases of noun compounds. In *Workshop on Semantic Evaluation (SemEval 2013)*, pages 138–143.
- Henrich, V. and Hinrichs, E. (2010). GernEdiT - The GermaNet Editing Tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.
- Henrich, V. and Hinrichs, E. W. (2011). Determining Immediate Constituents of Compounds in GermaNet. In *RANLP 2011*, pages 420–426.
- Heringer, H. J. (1984). Wortbildung: Sinn aus dem Chaos. *Deutsche Sprache*, 12:1–13.
- Hindle, D. and Rooth, M. (1993). Structural Ambiguity and Lexical Relations. *Computational Linguistics*.
- Hockenmaier, J. and Steedman, M. (2007). CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Comput. Linguist.*, 33(3):355–396.
- Hohenhaus, P. (1998). Non-lexicalizability as a characteristic feature of nonce word-formation in English and German. *Lexicology*, 2(4):237–80.
- Holz, F. and Biemann, C. (2008). Unsupervised and Knowledge-Free Learning of Compound Splits and Periphrases. In *CICLing 2008, Haifa: Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics, LNCS 4919*, pages 117–127. Springer LNCS.
- Hudson, R. A. (1975). Problems in the Analysis of Ed-Adjectives. *Journal of Linguistics*, 11(1):69–72.
- Humphreys, B. L., Lindberg, D. A., Schoolman, H. M., and Barnett, G. O. (1998). The Unified Medical Language System: an informatics research collaboration. *Journal of the American Medical Informatics Association : JAMIA*, 5(1):1–11.
- Huyssteen, G. B. V. and Zaanen, M. M. V. (2004). Learning Compound Boundaries for Afrikaans Spelling Checking. In *Proceedings of First Workshop on International Proofing Tools and Language Technologies*, pages 101–108.
- Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005). Bootstrapping Parsers via Syntactic Projection Across Parallel Texts. *Natural Language Engineering*.
- Iordachioaia, G. (2017). Deverbal Compounds between Nominalizations and Phrasal

## BIBLIOGRAPHY

- Compounds in English and Romanian. In V. Barbu Mititelu, M. I. and Iordachioaia, G., editors, *Lingvistica generala, lingvistica formala, lingvistica computationala. Omagiu profesorului Emil Ionescu la 60 de ani [General Linguistics, Formal Linguistics, Computational Linguistics. A Festschrift for Professor Emil Ionescu on his 60th Birthday]*, Bucharest. Bucharest University Press.
- Iordachioaia, G., van der Plas, L., and Jagfeld, G. (2016). The Grammar of English Deverbal Compounds and their Meaning. In *Proceedings of the Workshop on Grammar and Lexicon: interactions and interfaces (GramLex)*, pages 81–91, Osaka, Japan. The COLING 2016 Organizing Committee.
- Ivanova, M. and Wehrli, E. (2015). *Identification of Noun-Noun Compounds in the Context of Speech-to-Speech Translation*, pages 533–541. Springer International Publishing.
- Iwata, T., Mochihashi, D., and Sawada, H. (2010). Learning Common Grammar from Multilingual Corpus. In *ACL 2010*.
- Jagfeld, G., Ziering, P., and Van der Plas, L. (2017). Evaluating Compound Splitters Externally with Textual Entailment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada. Association for Computational Linguistics.
- Johnston, M. and Busa, F. (1999). Qualia structure and the compositional interpretation of compounds. In *E. Viegas (ed.), Breadth and depth of semantics lexicons*, pages 167–187. Dordrecht: Kluwer Academic.
- Jones, D. (1969). *An Outline of English Phonetics: 9th Ed.* W. Heffer & Sons Limited.
- Justeson, J. and Katz, S. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, pages 9–27.
- Karlsson, F. (1990). Constraint Grammar As a Framework for Parsing Running Text. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3, COLING '90*, pages 168–173.
- Kavuluru, R. and Harris, D. (2012). A Knowledge-Based Approach to Syntactic Disambiguation of Biomedical Noun Compounds. In *Proceedings of COLING 2012: Posters*, pages 559–568, Mumbai, India. The COLING 2012 Organizing Committee.
- Keller, F. and Lapata, M. (2003). Using the Web to Obtain Frequencies for Unseen Bigrams. *Computational Linguistics*, 29(3):459–484.
- Keller, F., Lapata, M., and Ourioupina, O. (2002). Using the Web to Overcome Data Sparseness. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02*, pages 230–237, Stroudsburg, PA, USA.

## BIBLIOGRAPHY

- Association for Computational Linguistics.
- Kim, S. N. and Baldwin, T. (2005). *Automatic Interpretation of Noun Compounds Using WordNet Similarity*, pages 945–956. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Kim, S. N. and Baldwin, T. (2006). Interpreting Semantic Relations in Noun Compounds via Verb Semantics. In *Proceedings of Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics*, pages 491–498, Sydney, Australia.
- Kim, S. N. and Baldwin, T. (2013). A Lexical Semantic Approach to Interpreting and Bracketing English Noun Compounds. *Natural Language Engineering*, 19(3).
- Kingdon, R. (1958). *The groundwork of English stress*. Longmans.
- Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*.
- Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *EACL*.
- Krenn, B. (2008). Description of evaluation resource – German PP-verb data. In *Proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, Marrakech, Morocco.
- Krott, A. (2001). *Analogy in Morphology: The Selection of Linking Elements in Dutch Compounds*. PhD thesis, Radboud University Nijmegen, Nijmegen.
- Krott, A. and Nicoladis, E. (2005). Large constituent families help children parse compounds. *Journal of Child Language*, 32:139–150.
- Krstev, C. (2008). *Processing of Serbian. Automata, Texts and Electronic Dictionaries*. Faculty of Philology of the University of Belgrade.
- Kulkarni, A. and Kumar, A. (2011). Statistical constituency parser for sanskrit compounds. In *Proceedings of ICON*.
- Kulkarni, A., Paul, S., Kulkarni, M., Kumar, A., and Surtani, N. (2012). Semantic Processing of Compounds in Indian Languages. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 1489–1502.
- Ladd, R. D. (1984). English compound stress. In *Intonation, Accent and Rhythm:*

## BIBLIOGRAPHY

- Studies in Discourse Phonology*. Berlin.
- Landis, J. R. and Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33.
- Langer, S. (1998). Zur Morphologie und Semantik von Nominalkomposita. In *KONVENS*.
- Langeweg, S. J. (1988). *The Stress System of Dutch*. Rijksuniversiteit te Leiden.
- Lapata, M. and Keller, F. (2004). The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based models for a range of NLP tasks. In *HLT-NAACL 2004*.
- Lapata, M. and Keller, F. (2005). Web-based Models for Natural Language Processing. 2(1).
- Lapata, M., Keller, F., and McDonald, S. (2001). Evaluating Smoothing Algorithms against Plausibility Judgments. In *The 39th Annual Meeting of the Association for Computational Linguistics and the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 346–353, Toulouse.
- Lapata, M. and Lascarides, A. (2003). Detecting Novel Compounds: The Role of Distributional Evidence. In *EACL*, pages 235–242. The Association for Computer Linguistics.
- Lapata, M., McDonald, S., and Keller, F. (1999). Determinants of Adjective-noun Plausibility. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 30–36, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Larrick, N. (1961). *Junior Science Book of Rain, Hail, Sleet and Snow*. Garrard Publishing Company.
- Larson, M., Willett, D., Köhler, J., and Rigoll, G. (2000). Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches.
- Lauer, M. (1994). Conceptual Association for Compound Noun Analysis. *CoRR*.
- Lauer, M. (1995a). Corpus Statistics Meet the Noun Compound: Some Empirical Results. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL, pages 47–54, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Lauer, M. (1995b). *Designing Statistical Language Learners: Experiments on Noun Compounds*. PhD thesis, Macquarie University.
- Lauer, M. and Dras, M. (1994). A Probabilistic Model of Compound Nouns. *CoRR*.
- Lazaridou, A., Vecchi, E. M., and Baroni, M. (2013). Fish transporters and miracle

## BIBLIOGRAPHY

- homes: How compositional distributional semantics can help NP parsing. In *In Proceedings of EMNLP*.
- Levi, J. N. (1978). *The Syntax and Semantics of Complex Nominals*. Academic Press.
- Liang, F. M. (1983). *Word hy-phen-a-tion by com-put-er*. PhD thesis, Stanford, CA, USA.
- Lieberman, M. and Sproat, R. (1992). The Stress and Structure of Modified Noun Phrases in English. In *Lexical Matters*, pages 131–181. Center for the Study of Language and Information.
- Lieber, R. (1992). *Deconstructing Morphology: Word Formation in Syntactic Theory*. University of Chicago Press.
- Lieber, R. (2009). I-E-Germanic: English. In Lieber, R. and Štekauer, P., editors, *The Oxford Handbook of Compounding*, chapter 18. Oxford University Press, Oxford.
- Lieber, R. and Štekauer, P. (2009). *The Oxford Handbook of Compounding*. Oxford Handbooks in Linguistics. OUP Oxford.
- Lin, D., Church, K. W., Ji, H., Sekine, S., Yarowsky, D., Bergsma, S., Patil, K., Pitler, E., Lathbury, R., Rao, V., Dalwani, K., and Narsale, S. (2010). New Tools for Web-Scale N-grams. In *LREC*. European Language Resources Association.
- Ljung, M. (1976). -Ed Adjectives Revisited. *Journal of Linguistics*, 12(1):159–168.
- Lowe, H. J. and Barnett, G. O. (1994). Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. *Jama*, 271(14).
- Macherey, K., Dai, A. M., Talbot, D., Popat, A. C., and Och, F. (2011). Language-independent Compound Splitting with Morphological Operations. In *ACL HLT 2011*.
- Macleod, C., Grishman, R., Meyers, A., Barrett, L., and Reeves, R. (1998). NOMLEX: A Lexicon of Nominalizations. In *In Proceedings of Euralex*, pages 187–193.
- Magnini, B., Zanolini, R., Dagan, I., Eichler, K., Neumann, G., Noh, T.-G., Pado, S., Stern, A., and Levy, O. (2014). The Excitement Open Platform for Textual Inferences. In *Proceedings of the ACL 2014 System Demonstrations*. ACL.
- Magnolini, S. and Magnini, B. (2015). Predicting Correlations Between Lexical Alignments and Semantic Inferences. In *RANLP*, pages 388–397. RANLP 2015 Organising Committee / ACL.
- Mansouri, A., Affendey, L. S., and Mamat, A. (2008). Named Entity Recognition Approaches. *IJCSNS International Journal of Computer Science and Network Security*, 8(2).
- Marchand, H. (1960). *The Categories and Types of Present-day English Word-formation: Synchronic-diachronic Approach*. Alabama linguistic & philological series. Otto Har-



## BIBLIOGRAPHY

- rassowitz.
- Marchand, H. (1967). Expansion, transposition, and derivation. *La Linguistique*, 3(1):13–26.
- Marchand, H. (1969). *The Categories and Types of Present-Day English Word-Formation: A Synchronic-Diachronic Approach*. Verlag C. H. Beck, Munich, 2nd edition.
- Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language*. MIT Press.
- Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*.
- Marek, T. (2006). Analysis of German Compounds using Weighted Finite State Transducers. Technical report.
- Mattens, W. H. M. (1970). *De Indifferentialis. Een Onderzoek naar het anumerieke Gebruik van het Substantief in het algemeen bruikbaar Nederlands*. PhD thesis, Assen: Van Gorcum.
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.
- Meibauer, J. (2003). Phrasenkomposita zwischen Wortsyntax und Lexikon. *Zeitschrift für Sprachwissenschaft : ZS*, 22(2):153–188.
- Melamed, I. D. (1997a). Automatic Discovery of Non-Compositional Compounds in Parallel Data. *CoRR*.
- Melamed, I. D. (1997b). Measuring Semantic Entropy. In *Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics*, pages 41–46.
- Ménard, P. A. and Barrière, C. (2014). Linked Open Data and Web Corpus Data for noun compound bracketing. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Team, T. G. B., Pickett, J. P., Holberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., and Aiden, E. L. (2010). Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*.
- Miller, G. A. (1995a). WordNet: a Lexical Database for English. *Communications of the ACM*.
- Miller, G. A. (1995b). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.
- Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics.

## BIBLIOGRAPHY

- Cognitive Science*, 34:1388–1429.
- Mithun, M. (1999). *The Languages of Native North America*. Cambridge Language Surveys. Cambridge University Press, New York.
- Moirón, B. V. and Tiedemann, J. (2006). Identifying Idiomatic Expressions using Automatic Word Alignment. In *Proceedings of the EACL 2006 Workshop on Multiword Expressions*.
- Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., O. Corazzari, Zampolli, A., Fanciulli, F., Massetani, M., Raffaelli, R., Basili, R., Pazienza, M. T., Saracino, D., Zanzotto, F., Mana, N., Pianesi, F., and Delmonte, R. (2000). The Italian Syntactic-Semantic Treebank: Architecture, Annotation, Tools and Evaluation. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 18–27, Centre Universitaire, Luxembourg. International Committee on Computational Linguistics.
- Monz, C. and de Rijke, M. (2001). *Shallow Morphological Analysis in Monolingual Information Retrieval for Dutch, German, and Italian*.
- Muischnek, K. and Kaalep, H. (2010). The variability of multi-word verbal expressions in Estonian. *Language Resources and Evaluation*, 44(1-2):115–135.
- Nagy T., I., Berend, G., and Vincze, V. (2011). Noun Compound and Named Entity Recognition and their Usability in Keyphrase Extraction. In *Recent Advances in Natural Language Processing, RANLP 2011, 12-14 September, 2011, Hissar, Bulgaria*, pages 162–169.
- Nagy T., I. and Vincze, V. (2013). *English Nominal Compound Detection with Wikipedia-Based Methods*, pages 225–232. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Nakov, P. (2013). On the Interpretation of Noun Compounds: Syntax, Semantics, and Entailment. *Natural Language Engineering*, 19(3):291–330.
- Nakov, P. and Hearst, M. (2005). Search Engine Statistics Beyond the N-gram: Application to Noun Compound Bracketing. In *CONLL 2005*.
- Nakov, P. I. (2007). *Using the Web as an Implicit Training Set: Application to Noun Compound Syntax and Semantics*. PhD thesis, EECS Department, University of California, Berkeley.
- Nastase, V. and Szpakowicz, S. (2003). Exploring noun-modifier semantic relations. In *Proceedings of the 5th International Workshop on Computational Semantics*.
- Navigli, R., Jurgens, D., and Vannella, D. (2013). SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Work-*

## BIBLIOGRAPHY

- shop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Navigli, R., Velardi, P., and Gangemi, A. (2003). Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, 18(1):22–31.
- Neef, M. (2009). I-E-Germanic: German. In Lieber, R. and Štekauer, P., editors, *The Oxford Handbook of Compounding*, chapter 20. Oxford University Press, Oxford.
- Neijt, A., Schreuder, R., and Jansen, C. (2010). Van boekenbonnen en feëverhale: De tussenklank e(n) in Nederlands en Afrikaanse samestellingen: vorm of betekenis? [The interfix e(n) in Dutch and Afrikaans compounds: form or meaning?]. *Nederlandse Taalkunde*, 15(2):125–147.
- Nicholson, J. and Baldwin, T. (2008). Interpreting Compound Nominalisations. In *Proceedings of the LREC 2008 Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 43–45, Marrakech, Morocco.
- Nießen, S. and Ney, H. (2000). Improving SMT quality with morpho-syntactic analysis. In *COLING 2000*, pages 1081–1085.
- Noh, T., Padó, S., Shwartz, V., Dagan, I., Nastase, V., Eichler, K., Kotlerman, L., and Adler, M. (2015). Multi-Level Alignments As An Extensible Representation Basis for Textual Entailment Algorithms. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, \*SEM 2015, June 4-5, 2015, Denver, Colorado, USA.*, pages 193–198.
- Nunberg, G., Sag, I. A., and Wasow, T. (1994). Idioms. In *Language*, pages 491–538. Cambridge University Press.
- Ó Séaghdha, D. (2007). Designing and Evaluating a Semantic Annotation Scheme for Compound Nouns. In *Proceedings of the 4th Corpus Linguistics Conference*.
- Ó Séaghdha, D. (2008). Learning Compound Noun Semantics. Technical Report UCAM-CL-TR-735, University of Cambridge, Computer Laboratory.
- Ó Séaghdha, D. (2008). *Learning compound noun semantics*. PhD thesis, Computer Laboratory, University of Cambridge. Published as University of Cambridge Computer Laboratory Technical Report 735.
- Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*.
- Ohta, T., Tateisi, Y., and Kim, J.-D. (2002). The GENIA Corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain. In *Proceedings of the Second International Conference on Human Language Technology Research, HLT '02*, pages 82–86, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

## BIBLIOGRAPHY

- Olive, J., Christianson, C., and McCary, J. (2011). *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer Publishing Company, Incorporated, 1st edition.
- Olsen, S. (2000). Compounding and stress in english: A closer look at the boundary between morphology and syntax. *Linguistische Berichte*, 181:55–70.
- Olsen, S. (2001). *Copulative compounds: a closer look at the interface between syntax and morphology*, pages 279–320. Springer Netherlands, Dordrecht.
- Ott, N. (2006). Evaluation of the BananaSplit Compound Splitter. Technical report.
- Padó, S., Noh, T.-G., Stern, A., Wang, R., and Zanolli, R. (2015). Design and realization of a modular architecture for textual entailment. *Natural Language Engineering*, 21(2):167–200.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Pecina, P. (2008). Reference Data for Czech Collocation Extraction. In *Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions*, pages 11–14, Marrakech, Morocco. ELRA.
- Pitler, E., Bergsma, S., Lin, D., and Church, K. W. (2010). Using Web-scale N-grams to Improve Base NP Parsing Performance. In *COLING 2010*.
- Plag, I. (2006). The variability of compound stress in english: structural, semantic and analogical factors. *English Language and Linguistics*, 10(1).
- Popović, M. and Ney, H. (2004). Towards the use of word stems and suffixes for statistical machine translation. In *Language Resources and Evaluation*, pages 1585–1588, Lisbon, Portugal.
- Popović, M., Stein, D., and Ney, H. (2006). Statistical Machine Translation of German Compound Words. In *FinTAL*, pages 616–624.
- Pustejovsky, J., Anick, P., and Bergler, S. (1993). Lexical Semantic Techniques for Corpus Analysis. *Comput. Linguist.*, 19(2):331–358.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Rackow, U., Dagan, I., and Schwall, U. (1992). Automatic translation of noun compounds. In *COLING 1992*, pages 1249–1253.
- Ramisch, C., de Medeiros Caseli, H., Villavicencio, A., Machado, A., and Finatto, M. J. (2010a). *A Hybrid Approach for Multiword Expression Identification*, pages 65–74.

## BIBLIOGRAPHY

- Springer Berlin Heidelberg, Berlin, Heidelberg.
- Ramisch, C., Villavicencio, A., and Boitet, C. (2010b). Multiword Expressions in the Wild?: The Mwetoolkit Comes in Handy. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, pages 57–60, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ramisch, C., Villavicencio, A., and Boitet, C. (2010c). mwetoolkit: a Framework for Multiword Expression Identification. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).
- Ramisch, C., Villavicencio, A., and Boitet, C. (2010d). Web-based and Combined Language Models: A Case Study on Noun Compound Identification. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1041–1049, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Reddy, S., McCarthy, D., and Manandhar, S. (2011). An Empirical Study on Compositionality in Compound Nouns. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP-11)*.
- Resnik, P. and Hearst, M. (1993). Structural ambiguity and conceptual relations. In *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pages 58 – 64.
- Resnik, P. and Smith, N. A. (2003). The Web As a Parallel Corpus. *Comput. Linguist.*, 29(3):349–380.
- Resnik, P. S. (1993). *Selection and Information: A Class-based Approach to Lexical Relationships*. PhD thesis, Philadelphia, PA, USA. UMI Order No. GAX94-13894.
- Riedl, M. and Biemann, C. (2016). Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods. In *NAACL-HTL 2016*.
- Roach, P. (1983). *English Phonetics and Phonology: A Practical Course*. Cambridge University Press.
- Rosario, B. and Hearst, M. (2001). Classifying the Semantic Relations in Noun Compounds via a Domain-Specific Lexical Hierarchy. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.
- Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). *Multiword Expressions: A Pain in the Neck for NLP*, pages 1–15.
- Salehi, B. and Cook, P. (2013). Predicting the Compositionality of Multiword Expressions Using Translations in Multiple Languages. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference*

## BIBLIOGRAPHY

- and the Shared Task: Semantic Textual Similarity*, pages 266–275, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Sampson, R. (1980). Stress in English N + N phrases: A further complicating factor. *English Studies*, 61(3):264–270.
- Savary, A. (2000). *Recensement et description des mots composés - méthodes et applications*. PhD thesis, Université de Marne la Vallée.
- Savary, A. and Piskorski, J. (2011). Language Resources for Named Entity Annotation in the National Corpus of Polish. *Control and Cybernetics*, 40(2):361–391.
- Scalise, S. (1984). *Generative morphology*. Studies in generative grammar. Foris Publications.
- Schiller, A. (2005). German Compound Analysis with wfsc. In *FSMNLP*, pages 239–246. Springer.
- Schlücker, B. and Hüning, M. (2010). Compounds and phrases. A functional comparison between German A + N compounds and corresponding phrases.
- Schmid, H. (1995). Improvements in Part-of-Speech Tagging with an Application to German. In *ACL SIGDAT-Workshop*.
- Schmid, H., Fitschen, A., and Heid, U. (2004). SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection. In *LREC 2004*, pages 1263–1266.
- Schneider, N., Danchik, E., Dyer, C., and Smith, N. (2014a). Discriminative Lexical Semantic Segmentation with Gaps: Running the MWE Gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.
- Schneider, N., Onuffer, S., Kazour, N., Danchik, E., Mordowanec, M. T., Conrad, H., and Smith, N. A. (2014b). Comprehensive Annotation of Multiword Expressions in a Social Web Corpus. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Schulte im Walde, S., Borgwaldt, S., and Jauch, R. (2012). Association Norms of German Noun Compounds. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).
- Schulte im Walde, S., Hättig, A., Bott, S., and Khvtisavrisvili, N. (2016). GhoSt-NN: A Representative Gold Standard of German Noun-Noun Compounds. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC*

## BIBLIOGRAPHY

- 2016), Paris, France. European Language Resources Association (ELRA).
- Schulte im Walde, S., Müller, S., and Roller, S. (2013). Exploring Vector Space Models to Predict the Compositionality of German Noun-Noun Compounds. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*.
- Schwarck, F., Fraser, A., and Schütze, H. (2010). Bitext-Based Resolution of German Subject-Object Ambiguities. In *NAACL-HLT 2010*.
- Schwartz, L., Aikawa, T., and Quirk, C. (2003). Disambiguation of English PP Attachment using Multilingual Aligned Data. In *MT Summit IX*.
- Selkirk, E. O. (1982). *The Syntax of Words*. MIT Press Cambridge, Mass.
- Shaoul, C. and Westbury, C. (2007). A USENET corpus (2005-2007). Technical report, University of Alberta.
- Smith, D. A. and Smith, N. A. (2004). Bilingual Parsing with Factored Estimation: Using English to Parse Korean. In *EMNLP 2004*.
- Snyder, B. and Barzilay, R. (2010). Climbing the Tower of Babel: Unsupervised Multilingual Learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 29–36.
- Snyder, B., Naseem, T., and Barzilay, R. (2009). Unsupervised Multilingual Grammar Induction. In *ACL-IJCNLP 2009*.
- Spencer, A. (1991). *Morphological Theory: An introduction to Word Structure in Generative Grammar*. Blackwell., Oxford.
- Spencer, A. (2003). Does english have productive compounding? In *Topics in Morphology: Selected Papers from the Third Mediterranean Morphology Meeting (Barcelona, September 20–22 2001)*, Barcelona. Institut Universitari de Lingüística Aplicada, Universitat Pompeu Fabra.
- Stymne, S. (2008). German Compounds in Factored Statistical Machine Translation. In *GoTAL*.
- Stymne, S. and Holmqvist, M. (2008). Processing of Swedish Compounds for Phrase-based Statistical Machine Translation. In *Proceedings of the 12th annual conference of the European Association for machine translation, EAMT '08*, pages 180–189.
- Sulubacak, U., Pamay, T., and Eryiğit, G. (2016). IMST: A Revisited Turkish Dependency Treebank. In *TurCLing 2016, The First International Conference on Turkic Computational Linguistics at CICLING 2016*, pages 1–6.
- Szarvas, G., Farkas, R., and Kocsor, A. (2006). *A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms*, pages 267–278.

## BIBLIOGRAPHY

- Springer Berlin Heidelberg, Berlin, Heidelberg.
- Szymanek, B. (1998). *Introduction to morphological analysis*. Warszawa: Panstwowe Wydawnictwo Naukowe.
- Tanaka, T. and Baldwin, T. (2003). Noun-Noun Compound Machine Translation: A Feasibility Study on Shallow Processing. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, MWE '03, pages 17–24.
- Thelen, M. and Riloff, E. (2002). A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 214–221, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *LREC 2012*.
- Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 Shared Task: Language-independent Named Entity Recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Trask, R. L. (1993). *A Dictionary of Grammatical Terms in Linguistics*. Linguistics (Routledge). Routledge.
- Tratz, S. and Hovy, E. (2010). A Taxonomy, Dataset, and Classifier for Automatic Noun Compound Interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 678–687, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Trommelen, M. and Zonneveld, W. (1986). Dutch morphology: Evidence for the right-hand head rule. *Linguistic Inquiry*, 17(1):147–169.
- Vadas, D. (2009). *Statistical Parsing of Noun Phrase Structure*. PhD thesis.
- Vadas, D. and Curran, J. (2007a). Adding Noun Phrase Structure to the Penn Treebank. In *ACL 2007*.
- Vadas, D. and Curran, J. R. (2007b). Large-scale Supervised Models for Noun Phrase Bracketing. In *PACLING 2007*.
- Vadas, D. and Curran, J. R. (2008). Parsing Noun Phrase Structure with CCG. In *ACL*



## BIBLIOGRAPHY

- 2008, *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 335–343.
- Valentin, K. and Bachmaier, H. (1995). *Sämtliche Werke in acht Bänden*, volume 3. Piper.
- Verhoeven, B., van Zaanen, M., Daelemans, W., and van Huyssteen, G. B. (2014). Automatic Compound Processing: Compound Splitting and Semantic Analysis for Afrikaans and Dutch. In *ComAComA 2014*, pages 20–30.
- Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A Model-theoretic Coreference Scoring Scheme. In *Proceedings of the 6th Conference on Message Understanding, MUC6 '95*, pages 45–52, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Vincze, V. and Csirik, J. (2010). Hungarian Corpus of Light Verb Constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010*, pages 1110–1118, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Vincze, V., Nagy T., I., and Berend, G. (2011). Multiword Expressions and Named Entities in the Wiki50 Corpus. In *RANLP*, pages 289–295. RANLP 2011 Organising Committee.
- Visch, E. (1990). *A Metrical Theory of Rhythmic Stress Phenomena*. Publications in language sciences. Foris Publications.
- Vlachos, A. (2011). Evaluating unsupervised learning for natural language processing tasks. In *Proceedings of the EMNLP Workshop on Unsupervised Learning in NLP*.
- Von der Heide, C. and Borgwaldt, S. (2009). Assoziationen zu Unter-, Basis- und Oberbegriffen. Eine explorative Studie. In *Proceedings of the 9th Norddeutsches Linguistisches Kolloquium*, pages 51–74.
- Warren, B. (1978). *Semantic Patterns of Noun-noun Compounds*. Acta Universitatis Gothoburgensis. Acta Universitatis Göthoburgensis.
- Wehrli, E. (1985). Design and Implementation of a Lexical Data Base. In *Proceedings of the Second Conference on European Chapter of the Association for Computational Linguistics, EACL '85*, pages 146–153, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Wehrli, E. (2007). Fips, a "Deep" Linguistic Multilingual Parser. In *Proceedings of the Workshop on Deep Linguistic Processing, DeepLP '07*, pages 120–127, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Weller, M., Cap, F., Müller, S., Schulte im Walde, S., and Fraser, A. (2014). Distin-

## BIBLIOGRAPHY

- guishing Degrees of Compositionality in Compound Splitting for Statistical Machine Translation. In *ComAComA 2014*.
- Weller, M. and Heid, U. (2012). Analyzing and Aligning German compound nouns. In *LREC 2012*.
- Widdows, D. (2008). Semantic Vector Products: Some Initial Investigations. In *Proceedings of the Second AAAI Symposium on Quantum Interaction*.
- Wiese, R. (1996). *The Phonology of German*. The phonology of the world's languages. Clarendon Press.
- Wisniewski, E. J. (1997). When concepts combine. *Psychonomic Bulletin & Review*, 4(2):167–183.
- Wuebker, J., Huck, M., Peitz, S., Nuhn, M., Freitag, M., Peter, J., Mansour, S., and Ney, H. (2012). Jane 2: Open Source Phrase-based and Hierarchical Statistical Machine Translation. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Demonstration Papers, 8-15 December 2012, Mumbai, India*, pages 483–492.
- Yang, M. and Kirchhoff, K. (2006). Phrase-based backoff models for machine translation of highly inflected languages. In *In Proceedings of EACL*, pages 41–48.
- Yang, Y. and Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97*, pages 412–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Yarowsky, D. and Ngai, G. (2001). Inducing Multilingual POS Taggers and NP Brackets via Robust Projection Across Aligned Corpora. In *NAACL 2001*.
- Yeh, A. (2000). More Accurate Tests for the Statistical Significance of Result Differences. In *COLING 2000*.
- Zeller, B. D. (2016). *Induction, Semantic Validation and Evaluation of a Derivational Morphology Lexicon for German*. PhD thesis, Heidelberg, Deutschland.
- Zeller, B. D., Snajder, J., and Padó, S. (2013). DERivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *ACL (1)*, pages 1201–1211. The Association for Computer Linguistics.
- Ziering, P., Müller, S., and Van der Plas, L. (2016). Top a Splitter: Using Distributional Semantics for Improving Compound Splitting. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 50–55, Berlin, Germany. Association for Computational Linguistics.
- Ziering, P. and Van der Plas, L. (2014). What good are 'Nominalkomposita' for 'noun

## BIBLIOGRAPHY

- compounds’: Multilingual Extraction and Structure Analysis of Nominal Compositions using Linguistic Restrictors. In *COLING 2014*.
- Ziering, P. and Van der Plas, L. (2015a). From a Distance: Using Cross-lingual Word Alignments for Noun Compound Bracketing. In *IWCS 2015*.
- Ziering, P. and Van der Plas, L. (2015b). One tree is not enough: Cross-lingual accumulative structure transfer for semantic indeterminacy. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 739–746, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.
- Ziering, P. and Van der Plas, L. (2016). Towards Unsupervised and Language-independent Compound Splitting using Inflectional Morphological Transformations. In *NAACL-HLT 2016*.
- Ziering, P., van der Plas, L., and Schütze, H. (2013a). Bootstrapping semantic lexicons for technical domains. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1321–1329, Nagoya, Japan. Asian Federation of Natural Language Processing.
- Ziering, P., van der Plas, L., and Schütze, H. (2013b). Multilingual Lexicon Bootstrapping - Improving a Lexicon Induction System Using a Parallel Corpus. In *IJCNLP 2013*.