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


Institut für Maschinelle Sprachverarbeitung
der Universität Stuttgart

2018
Erklärung (Statement of Authorship)

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., Fritzinger 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.
Abstract

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$_A$ rights$_B$ violation$_C$ being translated to German as Verletzung$_C$ der Menschenrechte$_A$$_B$, the fact that the constituent equivalent of violation$_C$, Verletzung$_C$, is separated from the other constituent equivalents, points us in the direction of a LEFT-branched structure, i.e., [human$_A$ rights$_B$] violation$_C$.

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 plural’ 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 theories, 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.


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.


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

Part A Preface 1

1 Introduction to the Thesis 3
  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

Part B Nature of Compounds 25

2 Introduction to Nature of Compounds 27
  2.1 Basic Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
## 3 General Aspects

### 3.1 Multi-Word Expressions

### 3.2 Naming Convention
- 3.2.1 Inconsistent Naming in Previous Literature
- 3.2.2 Naming in this Thesis

### 3.3 Productivity and Corpus Distribution

### 3.4 Functions of Compounds

### 3.5 Spelling
- 3.5.1 Closed Compounds
- 3.5.2 Hyphenated Compounds
- 3.5.3 Open Compounds
- 3.5.4 Mixed Spelling Forms

### 3.6 Constituents
- 3.6.1 Head
- 3.6.2 Headedness
- 3.6.3 Modifier
- 3.6.4 Complex Compounds and their Structure

### 3.7 Compound Classes
- 3.7.1 Universal Taxonomy
- 3.7.2 Neoclassical Compounds
- 3.7.3 Phrasal Compounds
- 3.7.4 Other Classes of Compounds
- 3.7.5 Compound Classes in this Thesis

### 3.8 Compound Semantics
- 3.8.1 Compositionality
- 3.8.2 Semantic Relation
- 3.8.3 Semantic Indeterminacy

### 3.9 Compounding across Languages
- 3.9.1 English
- 3.9.2 German
- 3.9.3 Dutch
- 3.9.4 Afrikaans

## 4 The Controversy of the Definition of Compounds

### 4.1 Various Ways of Compound Definition

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

### 4.3 Orthographical/Spelling Criteria
- 4.4.1 Word Inflection
- 4.4.2 Constituent Inflection

### 4.4 Morphological Criteria

### 4.5 Phonetic/prosodic criteria

### 4.6 Syntactic criteria
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 one .......... 68

4.7 Semantic criteria .................................................. 68
4.7.1 Permanence ....................................................... 68
4.7.2 Non-compositionality ........................................... 68
4.7.3 Lexicalization ..................................................... 69

5 Cross-lingual Observations of Compounds .......................... 71
5.1 Parallel Compounding ............................................. 71
5.1.1 Parallel Closed Compounding ................................ 73

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

5.2 Phrasal Translations ................................................ 74

6 Bottom Line of Nature of Compounds .................................. 81
6.1 Summary ............................................................ 81
6.2 Conclusion .......................................................... 82
6.3 Motivation and Outlook ............................................ 83

Part C Compound Identification ........................................... 85

7 Introduction to Compound Identification ............................. 87
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

8 Related Work on Compound Identification ............................ 93
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

9 Parallel Corpus .............................................. 119
  9.1 Language Selection in Europarl ............................. 120
  9.2 Preprocessing Steps ....................................... 120
  9.3 Opus Europarl ............................................. 121

10 Pilot Studies on Compound Identification .................... 123
  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

11 Compound Identification Method ............................ 147
  11.1 Compound Candidate Selection ............................ 147
  11.2 Multilingual Complementation ............................ 149
  11.3 Cross-lingual Validation ................................ 151

12 Europarl Nominal Compound Database ......................... 153
  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

13 Experiments on Compound Identification ..................... 159
  13.1 Setup ................................................... 159
  13.2 Results and Discussion ................................. 160

14 Bottom Line of Compound Identification ..................... 163
  14.1 Summary ............................................... 163
# Table of Contents

14.2 Conclusion ................................................. 164
14.3 Future Work ............................................. 166

## Part D Compound Splitting 169

### 15 Introduction to Compound Splitting 171
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

### 16 Related Work on Compound Splitting 179
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

### 17 Morphological Operation Patterns 195
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

### 18 Multilingual Compound Splitting 199
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

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

21 Bottom Line of Compound Splitting ............................................................................ 271
  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

Part E Compound Parsing .................................................................................................. 291

22 Introduction to Compound Parsing .............................................................................. 293
  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
# Table of Contents

22.3 Outline .................................................. 298

23 Related Work on Compound Parsing ............................................. 299

  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

24 Pilot Study using Aligned Phrase Patterns .................................... 325

  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

25 Compound Parsing Methods using Aligned Word Distance ..................... 335

  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

26 Bottom Line of the Compound Parsing ......................................... 363

  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

**Part F Bottom Line** 381

27 Summary, Conclusion and Future Work of the Thesis 383  
27.1 Summary of the Thesis .................................................. 383  
27.2 Conclusion of the Thesis ................................................ 385  
27.3 Future Work of the Thesis .............................................. 388

**Part G Appendix** 391

A Universal Surface Patterns 393  
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 German Constituent Inflection 395

C Split Point Format Compilation 397  
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

D Further Compound Splitting Gold Standards 405
  D.1 German Splitting Gold Standard of Cap (2014) ....................... 405
  D.2 Ghost-NN ......................................................... 406

E Annotation Guidelines for Creating the Europarl Nominal Compound-
  hood Ratings 407
  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

List of Abbreviations 413
List of Algorithms 421
List of Figures 425
List of Publications 429
List of Tables 431
List of Terms 435
Bibliography 457
Part A.

Preface
1. Introduction to the Thesis

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.

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. Introduction to the Thesis

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.

![Thematic structure of the thesis](image)

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 subtasks 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.

\[
\begin{array}{c|c|c|c}
& \text{Target compound} \\
\hline
\text{Hühnersuppenrezept} \downarrow & \text{determination of split points} \\
\hline
\text{Hühner suppen rezept} \downarrow \downarrow \downarrow & \text{constituent forms} \\
\hline
\text{Huhn Suppe Rezept} & \text{constituent lemmas} \\
\end{array}
\]

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 [natural language 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.
1. Introduction to the Thesis

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
1. Introduction to the Thesis

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 | Arten* ‘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 *Gastrum* 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., [[natural language processing] vs. [natural 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.
1. Introduction to the Thesis

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* ‘recuperated 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 subcategorial 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’)).

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, Fritzinger 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 Fritzinger 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
1. Introduction to the Thesis

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 Ψ, 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.
1. Introduction to the Thesis

**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. Introduction to the Thesis

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., Fritzinger 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.
1. Introduction to the Thesis

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\(^1\). 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).

\(^1\)statmt.org/europarl
1. Introduction to the Thesis

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 compound- hood 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.
1. Introduction to the Thesis

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.
1. Introduction to the Thesis

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\(^1\).

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

\(^1\)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 discontiguously 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).
3. General Aspects

**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 (Mansourri 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).
3. General Aspects

Homogenous compositions\(^1\) of a word category Θ is called ‘Θ 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 (\(k\)NC), 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 (\(k\)C).

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

---

\(^1\)Compositions include both closed compounds and open sequences.
3. General Aspects

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 Shim bun 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. General Aspects

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 neologistic 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 → Schreibheft ‘to write + book → writing book’), word-internal vowel adaption (e.g., the German Umlautung as in Mutter + Rente → Mütterrente ‘mother + pension → mother’s pension’) or word-final suffixation (i.e., adding so-called linking elements as in the German Kind + Lied → Kinderlied ‘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 decomposing (as done with closed compounds, see the compound splitting Part D), because the hyphens already indicate the split points, i.e.,
3. General Aspects

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).
3. General Aspects

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

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.

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.

\[
Cat_n = \frac{(2n)!}{(n+1)! \cdot n!}
\]  

(3.1)
3. General Aspects

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

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
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., cutthroat). 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 “metonomy 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., \textit{the man is eyed} vs. \textit{the man is blue-eyed}). While being considered as a suffixed exocentric compound (i.e., [\textit{grey-beard}]+\textit{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., \textit{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., \textit{poet painter} refers to an entity being both \textit{poet and painter}, or \textit{singer songwriter} refers to an entity being both \textit{singer and songwriter}. As discussed by Lieber (2009), coordinate endocentric compounds are not common in English. Examples of this class include \textit{spiderman}, \textit{comedy drama} or \textit{king emperor} for nouns, \textit{blue green} and \textit{deaf mute} for adjectives and \textit{trickle irrigate} or \textit{slam dunk} for verbs. A more productive class are 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 \textit{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., \textit{anthropology} composed of \textit{anthropo ‘human’} and \textit{logy ‘science’}, i.e., the science of the humans. Bisetto and Scalise (2005) categorized neoclassical compounds as subordinate compounds, e.g., \textit{hydrophobia} (\textit{hydro ‘water’} + \textit{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: \textit{Konfixkomposita}) are also a phenomenon in the German language, e.g., \textit{Thermo|stat ‘thermostat’}. 

42
3. General Aspects

compounds

subordinate
- endocentric
  - apple cake
  - brain death
  - finger print
  - mail man
  - sun glasses
  - water pipe
  - taxi driver
  - stone cutter
  - arm control
  - baby care
  - agora phobia
- exocentric
  - kill joy
  - cutthroat

attributive
- endocentric
  - blue cheese
  - atomic bomb
  - back yard
  - French kiss
  - ape man
  - ghost writer
  - key word
  - public opinion
  - sword fish
  - snail mail
- exocentric
  - white collar
  - green house
  - pale face
  - long legs
  - free lance

coordinate
- endocentric
  - actor author
  - priest hermit
  - singer bassist
  - dancer singer
  - artist designer
  - fighter bomber
  - king emperor
  - merchant tailor
  - poet painter
- exocentric
  - mind brain
  - mother child
  - north east

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*\(^2\) (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’.

\(^2\)en.wikipedia.org/wiki/Austria-Hungary
3. General Aspects

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.

![Tree structure for 'Natural Language Processing'](image)

**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: \( w_1 \sim w_2 = \text{the policy of the government} \); \( w_1 \sim w_3 = \text{the decisions made by the government} \); \( w_2 \sim w_3 = \text{the decisions about the policy} \).

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 ($k$Cs):

\[
\text{a } kC \text{ is semantically indeterminate between two structures } \alpha \text{ and } \beta \text{ if situations that verify the meaning associated with } \alpha \text{ also make the meaning associated with } \beta \text{ true}
\]

Dealing with Semantic Indeterminacy

Despite the fact that previous work discussed semantic indeterminacy of $k$Cs, 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 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.

---

3George 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. General Aspects

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

<table>
<thead>
<tr>
<th>PoS Combination</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nominal Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + NOUN</td>
<td>hot dog</td>
</tr>
<tr>
<td>NOUN + NOUN</td>
<td>database</td>
</tr>
<tr>
<td>PREP + NOUN</td>
<td>underwater</td>
</tr>
<tr>
<td>VERB + NOUN</td>
<td>cutthroat</td>
</tr>
<tr>
<td><strong>Verbal Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + VERB</td>
<td>to highlight</td>
</tr>
<tr>
<td>NOUN + VERB</td>
<td>to fingerprint</td>
</tr>
<tr>
<td>PREP + VERB</td>
<td>to withdraw</td>
</tr>
<tr>
<td>VERB + VERB</td>
<td>to freeze-dry</td>
</tr>
<tr>
<td><strong>Adjectival Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + ADJ</td>
<td>dark-blue</td>
</tr>
<tr>
<td>NOUN + ADJ</td>
<td>bulletproof</td>
</tr>
<tr>
<td>PREP + ADJ</td>
<td>over-eager</td>
</tr>
<tr>
<td><strong>Adverbial Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + ADV</td>
<td>left-most</td>
</tr>
<tr>
<td>NOUN + ADV</td>
<td>headfirst</td>
</tr>
<tr>
<td>VERB + ADV</td>
<td>countdown</td>
</tr>
<tr>
<td><strong>Prepositional Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + PREP</td>
<td>forthwith</td>
</tr>
<tr>
<td>NOUN + PREP</td>
<td>timeout</td>
</tr>
<tr>
<td>PREP + PREP</td>
<td>without</td>
</tr>
<tr>
<td>VERB + PREP</td>
<td>cut-off</td>
</tr>
</tbody>
</table>

Table 3.3.: Possible PoS combinations for English binary compounds

There are various PoS combinations of English compounds, some of which are listed
3. General Aspects

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’.
3. General Aspects

Word Categories of the Head

<table>
<thead>
<tr>
<th>PoS Combination</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nominal Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + NOUN</td>
<td>Großbaustelle ‘large construction site’</td>
</tr>
<tr>
<td>NOUN + NOUN</td>
<td>Fußball ‘soccer’</td>
</tr>
<tr>
<td>PREP + NOUN</td>
<td>Überssee ‘overseas’</td>
</tr>
<tr>
<td>VERB + NOUN</td>
<td>Zeigefinger ‘forefinger’</td>
</tr>
<tr>
<td><strong>Verbal Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + VERB</td>
<td>kleinschlagen ‘to knock to pieces’</td>
</tr>
<tr>
<td>NOUN + VERB</td>
<td>eislaufen ‘to ice-skate’</td>
</tr>
<tr>
<td>PREP + VERB</td>
<td>unterlassen ‘to refrain from’</td>
</tr>
<tr>
<td>VERB + VERB</td>
<td>kennenlernen ‘to get to known’</td>
</tr>
<tr>
<td><strong>Adjectival Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + ADJ</td>
<td>bestmöglich ‘best possible’</td>
</tr>
<tr>
<td>NOUN + ADJ</td>
<td>hundemüde ‘dog-tired’</td>
</tr>
<tr>
<td>PREP + ADJ</td>
<td>unterernährt ‘undernourished’</td>
</tr>
<tr>
<td>VERB + ADJ</td>
<td>treffsicher ‘accurate’</td>
</tr>
<tr>
<td><strong>Adverbial Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>ADJ + ADV</td>
<td>schlechthin ‘plainly’</td>
</tr>
<tr>
<td><strong>Prepositional Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>NOUN + PREP</td>
<td>kopfüber ‘headfirst’</td>
</tr>
</tbody>
</table>

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 Zinses|zins ‘compound interest’ (Günter, 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 abbre-
3. General Aspects

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 characteristic of German compounding is constituent inflection, i.e., the addition of suffixes (i.e., Fugen|lemente ‘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): Kapitän|-(mützen) und Admiral|mützen ‘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., Liebe|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|pflege ‘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’.

![Structural ambiguity in German with different primary stress](image)

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

<table>
<thead>
<tr>
<th>PoS Combination</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nominal Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>NOUN + NOUN</td>
<td>vleessoep ‘meat soup’</td>
</tr>
<tr>
<td>VERB + NOUN</td>
<td>speelveld ‘play field’</td>
</tr>
<tr>
<td>ADJ + NOUN</td>
<td>sneltrein ‘express train’</td>
</tr>
<tr>
<td><strong>Verbal Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>PREP + VERB</td>
<td>opbellen ‘to phone’</td>
</tr>
<tr>
<td>NOUN + VERB</td>
<td>rangschikken ‘to sort’</td>
</tr>
<tr>
<td><strong>Adjectival Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>NOUN + ADJ</td>
<td>vrouwvriendelijk ‘woman friendly’</td>
</tr>
<tr>
<td>VERB + ADJ</td>
<td>fluisterzacht ‘whisper soft’</td>
</tr>
<tr>
<td>ADJ + ADJ</td>
<td>donkerblond ‘dark blond’</td>
</tr>
<tr>
<td><strong>Prepositional Compounds</strong></td>
<td></td>
</tr>
<tr>
<td>PREP + PREP</td>
<td>voorover ‘headfirst’</td>
</tr>
</tbody>
</table>

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 adjectival 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).
3. General Aspects

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 ‘weather forecast’, where the modifier weers ‘weather’ 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 hond|e|hok ‘doghouse’. Usually, the additive suffixes used in Dutch constituent inflection are: $+$s, $+$e, $+$en and $+$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 deskundigen|congres ‘experts conference’, steden|raad ‘cities council’ or boeken|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’))
3. General Aspects

**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 *kisfabriek* ‘coffin factories’) or word derivation (e.g., *katkosagtig* ‘like cat food’).

Linking elements joining modifier and head in Afrikaans compounds (as in *besigheid|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.

<table>
<thead>
<tr>
<th>Linking element</th>
<th>Example</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>+s</td>
<td>verbinding</td>
<td>s</td>
</tr>
<tr>
<td>+e</td>
<td>hond</td>
<td>e</td>
</tr>
<tr>
<td>+ns</td>
<td>lewens</td>
<td>drang ‘life force’</td>
</tr>
<tr>
<td>+er</td>
<td>kinder</td>
<td>skoen ‘children shoes’</td>
</tr>
<tr>
<td>+ens</td>
<td>nooier</td>
<td>van ‘maiden name’</td>
</tr>
<tr>
<td>+n</td>
<td>buiten</td>
<td>gewone ‘extraordinary’</td>
</tr>
<tr>
<td>hyphen</td>
<td>sterre-energie ‘star power’</td>
<td>656 (9.02%)</td>
</tr>
</tbody>
</table>

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’.

<table>
<thead>
<tr>
<th>Compound size</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>31,358 (78.30%)</td>
</tr>
<tr>
<td>3</td>
<td>7993 (19.96%)</td>
</tr>
<tr>
<td>4</td>
<td>663 (1.66%)</td>
</tr>
<tr>
<td>5</td>
<td>35 (0.09%)</td>
</tr>
<tr>
<td>6</td>
<td>2 (0.005%)</td>
</tr>
</tbody>
</table>

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.

The largest compound Huyssteen and Zaanen (2004) observed was

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

\[4\text{A similar distribution of German, Dutch and Afrikaans compound size is presented in Chapter 18, Tables 18.5, 18.7 and 18.9.}\]
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.

Marchard (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
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
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\(^1\) 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.

\[(5)\]
\[
\begin{align*}
\text{a. } & \text{The plane overflew the field} \\
\text{b. } & \text{*The plane flew the field} \\
\text{c. } & \text{The editor proofreads the article} \\
\text{d. } & \text{The editor reads the article}
\end{align*}
\]

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*\(^1\)

---

\(^{1}\) The *lexeme* is defined as a set of syntactic and semantic features shared by one or several morphosyntactic 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: *Alt*papier ‘scrap paper’ vs. *altes* Papier ‘old paper’, *Jung*frauen ‘virgins’ vs. *junge* Frauen ‘young women’ or *best*bezahlter 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 *x*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
4. The Controversy of the Definition of Compounds

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 ο, 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).
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 Google² 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*
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

<table>
<thead>
<tr>
<th>English</th>
<th>Danish</th>
<th>German</th>
<th>Dutch</th>
<th>Swedish</th>
</tr>
</thead>
<tbody>
<tr>
<td>climate change</td>
<td>klimaændringerne</td>
<td>Klimawandel</td>
<td>klimaatverandering</td>
<td>klimaförändringarna</td>
</tr>
<tr>
<td>nuclear power</td>
<td>kernekraft</td>
<td>Kernenergie</td>
<td>kernenergie</td>
<td>kärnkraft</td>
</tr>
<tr>
<td>action plan</td>
<td>handlingsplan</td>
<td>Aktionsplan</td>
<td>actieplan</td>
<td>handlingsplan</td>
</tr>
<tr>
<td>euro area</td>
<td>euroområdet</td>
<td>Euroraum</td>
<td>eurozone</td>
<td>euroområdet</td>
</tr>
<tr>
<td>energy efficiency</td>
<td>energieffektivitet</td>
<td>Energieeffizienz</td>
<td>energie-efficiëntie</td>
<td>energieffektivitet</td>
</tr>
<tr>
<td>free trade</td>
<td>frihandel</td>
<td>Freihandel</td>
<td>vrijhandel</td>
<td>frihandel</td>
</tr>
</tbody>
</table>

Table 5.1.: Examples of parallel compounding

Our first observation 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,
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.

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

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.

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

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.

<table>
<thead>
<tr>
<th>English</th>
<th>Danish</th>
<th>German</th>
<th>Dutch</th>
<th>Swedish</th>
</tr>
</thead>
<tbody>
<tr>
<td>same time</td>
<td>17.0%</td>
<td>21.2%</td>
<td>44.7%</td>
<td>70.5%</td>
</tr>
<tr>
<td>last year</td>
<td>1.6%</td>
<td>2.3%</td>
<td>0.8%</td>
<td>8.3%</td>
</tr>
<tr>
<td>great deal</td>
<td>18.9%</td>
<td>35.4%</td>
<td>11.2%</td>
<td>4.7%</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>English</th>
<th>Equivalents in other European languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>airport</td>
<td><strong>Danish</strong></td>
</tr>
<tr>
<td></td>
<td><em>lufthavnen</em></td>
</tr>
<tr>
<td></td>
<td><em>Spanish</em></td>
</tr>
<tr>
<td></td>
<td><em>aeropuerto</em></td>
</tr>
<tr>
<td>motorbikes</td>
<td><strong>Danish</strong></td>
</tr>
<tr>
<td></td>
<td><em>motorcykler</em></td>
</tr>
<tr>
<td></td>
<td><strong>Spanish</strong></td>
</tr>
<tr>
<td></td>
<td><em>motocicletas</em></td>
</tr>
<tr>
<td>microphone</td>
<td><strong>Danish</strong></td>
</tr>
<tr>
<td></td>
<td><em>mikrofon</em></td>
</tr>
<tr>
<td></td>
<td><strong>Spanish</strong></td>
</tr>
<tr>
<td></td>
<td><em>micrófono</em></td>
</tr>
<tr>
<td>ecosystems</td>
<td><strong>Danish</strong></td>
</tr>
<tr>
<td></td>
<td><em>økosystemer</em></td>
</tr>
<tr>
<td></td>
<td><strong>Spanish</strong></td>
</tr>
<tr>
<td></td>
<td><em>ecosistemas</em></td>
</tr>
<tr>
<td>spokesperson</td>
<td><strong>Danish</strong></td>
</tr>
<tr>
<td></td>
<td><em>talsmand</em></td>
</tr>
<tr>
<td></td>
<td><strong>Spanish</strong></td>
</tr>
<tr>
<td></td>
<td><em>portavoz</em></td>
</tr>
</tbody>
</table>

Table 5.5.: Examples for parallel closed compounding

73
5. Cross-lingual Observations of Compounds

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.

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

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.

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

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
5. Cross-lingual Observations of Compounds

Entwicklung, the 3NC trade defence instruments can be translated to Swedish as handelspolitiska skyldsinstrumenten 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.

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

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
5. Cross-lingual Observations of Compounds

(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 Equivalents

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 an 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-
5. Cross-lingual Observations of Compounds

spective word alignments: this phrasal equivalent points to a balanced tree structure, shown in Figure 5.1.

\[
\text{energy efficiency action plans} \\
\text{Aktionspläne für Energieeffizienz}
\]

\[
\text{energy efficiency action plans} \\
\text{Energieeffizienz Aktionspläne}
\]

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 boekenprijsregelingen (lit: ‘fixed book price schemes’) and in French as règlements concernant les prix fixes du livre (lit: ‘schemes concerning the fixed price of books’).
5. Cross-lingual Observations of Compounds

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.

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
<th>made_of?</th>
</tr>
</thead>
<tbody>
<tr>
<td>glass bottle</td>
<td>bouteille en verre</td>
<td>✓</td>
</tr>
<tr>
<td>glass jar</td>
<td>pot en verre</td>
<td>✓</td>
</tr>
<tr>
<td>leather jacket</td>
<td>vest en cuir</td>
<td>✓</td>
</tr>
<tr>
<td>paper basket</td>
<td>corbeille à papier</td>
<td>X</td>
</tr>
<tr>
<td>Iron curtain</td>
<td>rideau de fer</td>
<td>X</td>
</tr>
</tbody>
</table>

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 multilingual 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?
7. Introduction to Compound Identification

**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. Introduction to 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 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.

1 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.

\[
C = \{(w_2, w_3)|w_1w_2w_3w_4; w_1, w_4 \notin N; w_2, w_4 \in N\}
\] (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.
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 Google1 and Yahoo!2. 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

---

1 google.com
2 yahoo.com
8. Related Work on Compound Identification

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. Related Work on Compound Identification

8.1.5. General MWEs

Instead of focusing 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\textsubscript{trial} (involving a NCC candidate) and TM\textsubscript{basic} (without NCC), the NCC candidate is considered as valid if the value of TM\textsubscript{trial}’s objective function is higher than the value for TM\textsubscript{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\textsuperscript{3}. From these parses, 191K tuples of main verb and PP argument were collected. Using a combination

\textsuperscript{3}http://www.let.rug.nl/~vannoord/alp/Alpino
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. Therefor, 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 $S = s_1 \ldots s_n$ (with $n \geq 2$) is aligned to any (English) target word sequence $T = t_1 \ldots t_m$ (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
8. Related Work on Compound Identification

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\(^4\) 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 discontiguous MWEs, whereas the statistical approach is designed for expressions containing contiguous words.

\(^4\)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-ood 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 \( L_{\text{term}} \) (i.e., the language that specifies the set of candidate terms for all languages). The language \( L_{\text{term}} \) should be a closed compounding language such as German. Furthermore, German is an ideal candidate for \( L_{\text{term}} \), 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, \( \{L_{\text{term}}, L_i\} \), a term in \( L_i \) is defined as a word sequence that is aligned to a term in \( L_{\text{term}} \).

For example, the English word sequence \textit{liquid phase hydrogenation} is classified as term, because it is aligned to the German nominal compound \textit{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).

\[
(\text{ADJ|GERUND|NOUN})^* \text{ NOUN (PREP NOUN^+)}? \tag{8.2}
\]

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: (ADJ|NOUN)* 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).

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

(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.
8. Related Work on Compound Identification

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\(^5\), 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.,

\(^5\)morbocomp.sslmit.unibo.it
8. Related Work on Compound Identification

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

<table>
<thead>
<tr>
<th>Basque</th>
<th>Bulgarian</th>
<th>Byelorussian</th>
<th>Catalan</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dutch ✓</td>
<td>English ✓</td>
<td>Finnish ✓</td>
<td>French ✓</td>
<td>German ✓</td>
</tr>
<tr>
<td>Greek ✓</td>
<td>Hebrew ✓</td>
<td>Hungarian ✓</td>
<td>Italian ✓</td>
<td>Japanese</td>
</tr>
<tr>
<td>Korean</td>
<td>Latin ✓</td>
<td>Norwegian ✓</td>
<td>Polish ✓</td>
<td>Portuguese ✓</td>
</tr>
<tr>
<td>Russian</td>
<td>Serbo-croatian ✓</td>
<td>Spanish ✓</td>
<td>Swedish ✓</td>
<td>Turkish</td>
</tr>
</tbody>
</table>

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.
8. Related Work on Compound Identification

### Information field in MORBO/COMP

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

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

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

<table>
<thead>
<tr>
<th>Relation class</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>191 (9.6%)</td>
</tr>
<tr>
<td>HAVE</td>
<td>199 (10%)</td>
</tr>
<tr>
<td>IN</td>
<td>308 (15.4%)</td>
</tr>
<tr>
<td>ACTOR</td>
<td>236 (11.8%)</td>
</tr>
<tr>
<td>INST</td>
<td>266 (13.3%)</td>
</tr>
<tr>
<td>ABOUT</td>
<td>243 (12.2%)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1443 (100%)</strong></td>
</tr>
</tbody>
</table>

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

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\(^6\). 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 \(<\text{NP}_{EN}, \text{NP}_{ES}, \text{NP}_{IT}, \text{NP}_{FR}, \text{NP}_{PT}, \text{NP}_{RO}, \text{semantic relation}>>\), for example, \(<\text{development cooperation}; \text{cooperación para el desarrollo}; \text{cooperação allo sviluppo}; \text{coopération au développement}; \text{cooperare pentru dezvoltare}; \text{PURPOSE/FOR}>>\).

The BNC Compound Nominalization Dataset, created by Nicholson and

\(^6\)CLUVI - Linguistic Corpus of the University of Vigo - Parallel Corpus 2.1 - http://sli.uvigo.es/CLUVI/
8. Related Work on Compound Identification

Baldwin (2008), contains English 2NCs and focuses on a deeper analysis for the sub-group 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.

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

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 (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.

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

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-
human 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 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”.

<table>
<thead>
<tr>
<th>Compound</th>
<th>Meaningfulness</th>
<th>Inverse</th>
<th>Meaningfulness</th>
</tr>
</thead>
<tbody>
<tr>
<td>school pony</td>
<td>2.5</td>
<td>pony school</td>
<td>4</td>
</tr>
<tr>
<td>tomato juice</td>
<td>4</td>
<td>juice tomato</td>
<td>4</td>
</tr>
<tr>
<td>prison door</td>
<td>4</td>
<td>door prison</td>
<td>2</td>
</tr>
<tr>
<td>child seat</td>
<td>4</td>
<td>seat child</td>
<td>1</td>
</tr>
<tr>
<td>wine bottle</td>
<td>4</td>
<td>bottle wine</td>
<td>1.5</td>
</tr>
</tbody>
</table>

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. 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)
(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).

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

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
8. Related Work on Compound Identification

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.

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

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 of the oldest methods is called the multiple tube method. 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.
(1) My wife had taken her '07 Ford Fusion in for a routine oil change.

(2) he was willing to budge a little on the price which means a lot to me.

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

Additional MWE Resources

The project about PARSin and Multi-word Expressions (PARSEME)\(^9\) 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\(^10\) 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\(^11\), the Dictionary of Neologisms in Bulgarian Language\(^12\), the British English Source Lexicon (BESL)\(^13\), the Italian Syntactic-Semantic Treebank (ISST) (Montemagni et al., 2000); the Oxford English phonetics files\(^14\) 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 Doddington et al. (2004), Tjong Kim Sang (2002), Tjong Kim Sang and De Meulder (2003), Grishman and Sundheim (1995) and Chinchor (1998).

\(^9\)typo.uni-konstanz.de/parseme
\(^10\)clarino.uib.no/iness/page?page-id=MWEs_in_Parseme
\(^11\)http://metashare.ilsp.gr:8080/repository/browse/bulgarian-sense-annotated-corpus/b7d5478666cd11e281b65cf3fc88b705fc4e099156a4a94997974778d015eaa8/
\(^12\)http://metashare.ilsp.gr:8080/repository/browse/dictionary-of-neologisms-in-bulgarian-language/7ad446f268ad11e281b65cf3fc88b70dd4a3a216cb34a998c25fda3d4e70b2a
\(^13\)http://metashare.ilsp.gr:8080/repository/browse/british-english-source-lexicon-besl-version-22/dc410e62d681c2b1e40025901f6eaf8112b519c346f8a10378af93ecce2a
\(^14\)http://metashare.ilsp.gr:8080/repository/browse/oxford-english-phonetics-files/e986bb8ede6911e2b1e40025901f6eacf808bda74be4dc4879f8d2cf624cc4a
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\(^1\)), 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\(^2\) to one of the most well-known parallel corpora for computational linguistics, EUROPARL\(^3\). 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.

---

\(^1\)wikipedia.org
\(^2\)Nevertheless, we expect to see similar results for other parallel corpora.
\(^3\)statmt.org/europarl
9. Parallel Corpus

9.1. Language Selection in Europarl

All cross-lingual discussions, methods and experiments in this thesis are based on the 7th 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), Finni-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 (≈ 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.

![Language Selection Table]

Table 9.1.: EUROPARL language selection

9.2. Preprocessing Steps

For most methods and experiments in this thesis, the parallel corpus needs to be preprocessed. This step includes tokenization, PoS tagging and lemmatization.
9. Parallel Corpus

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 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 (Schmid, 1995) has been used.

For PoS tagging the Danish, Portuguese and Swedish part, the Hunpos tagger has been used.

OPUS does not provide PoS and lemma information for Greek. Therefor, we used a Greek model of the MATE tagger for preprocessing the Greek part of EUROPARL.

\[\text{Source: opus.lingfil.uu.se, cis.uni-muenchen.de/~schmid/tools/TreeTagger, code.google.com/p/hunpos/downloads/list, code.google.com/p/mate-tools}\\]
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\(^1\) Calc, a free and open-source spreadsheet program.

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.

\(^1\)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.

![Figure 10.2.: Examples of spreadsheet-based nominal compound annotation](image)

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)
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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sentences</th>
<th>Compound tokens</th>
<th>Avg. tokens / sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation1</td>
<td>270</td>
<td>624</td>
<td>2.31</td>
</tr>
<tr>
<td>Annotation2</td>
<td>195</td>
<td>346</td>
<td>1.77</td>
</tr>
<tr>
<td>Combination</td>
<td>394</td>
<td>824</td>
<td>2.09</td>
</tr>
</tbody>
</table>

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
10. Pilot Studies on Compound Identification

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.

\[
Jaccard = \frac{|Anno_1 \cap Anno_2|}{|Anno_1 \cup Anno_2|}
\]  
(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.

<table>
<thead>
<tr>
<th>Iter</th>
<th># extractions (\cap) Anno2</th>
<th>Jaccard</th>
<th>Precision</th>
<th>Recall</th>
<th>F(_1)-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>38</td>
<td>0.389</td>
<td>0.568</td>
<td>0.553</td>
<td>0.560</td>
</tr>
<tr>
<td>2</td>
<td>39</td>
<td>0.528</td>
<td>0.667</td>
<td>0.718</td>
<td>0.691</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>0.717</td>
<td>0.826</td>
<td>0.844</td>
<td>0.835</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th># extractions (\cap) Anno2</th>
<th>Jaccard</th>
<th>Precision</th>
<th>Recall</th>
<th>F(_1)-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>119</td>
<td>0.431</td>
<td>0.660</td>
<td>0.555</td>
<td>0.603</td>
</tr>
</tbody>
</table>

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.
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.

<table>
<thead>
<tr>
<th>Anno1</th>
<th>Anno2</th>
</tr>
</thead>
<tbody>
<tr>
<td>specific aid</td>
<td>USA</td>
</tr>
<tr>
<td>economic policy</td>
<td>international</td>
</tr>
<tr>
<td>political forces</td>
<td>a third</td>
</tr>
<tr>
<td>statistical data</td>
<td>ourselves</td>
</tr>
<tr>
<td>daily crime</td>
<td>PSE</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Iter</th>
<th>Average rating of Compoundhood</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anno1</td>
<td>Anno2</td>
<td>Difference</td>
</tr>
<tr>
<td>1</td>
<td>2.571</td>
<td>1.714</td>
<td>0.857</td>
</tr>
<tr>
<td>2</td>
<td>2.571</td>
<td>1.750</td>
<td>0.821</td>
</tr>
<tr>
<td>3</td>
<td>2.316</td>
<td>1.763</td>
<td>0.553</td>
</tr>
</tbody>
</table>

Table 10.5.: Average difference in compoundhood rating in the training stage
For the training stage, the compoundhood rating agreement is given in Table 10.5, and for the agreement set in Table 10.6.

<table>
<thead>
<tr>
<th>Average rating of Compoundhood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anno1</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>2.303</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Iter</th>
<th>Average rating of LC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anno1</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>Spelling</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.667</td>
</tr>
<tr>
<td>2</td>
<td>1.107</td>
</tr>
<tr>
<td>3</td>
<td>1.132</td>
</tr>
<tr>
<td>Inseparability</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.429</td>
</tr>
<tr>
<td>2</td>
<td>2.393</td>
</tr>
<tr>
<td>3</td>
<td>2.289</td>
</tr>
<tr>
<td>Inability to modify the modifier</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.095</td>
</tr>
<tr>
<td>2</td>
<td>1.964</td>
</tr>
<tr>
<td>3</td>
<td>1.868</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Iter</th>
<th>Average rating of LC</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anno1</td>
<td>Anno2</td>
</tr>
<tr>
<td>Inability to replace the head by one</td>
<td>2.286</td>
<td>2.048</td>
</tr>
<tr>
<td></td>
<td>2.357</td>
<td>2.214</td>
</tr>
<tr>
<td></td>
<td>2.526</td>
<td>2.368</td>
</tr>
<tr>
<td>Inflection of the modifier</td>
<td>2.286</td>
<td>2.048</td>
</tr>
<tr>
<td></td>
<td>2.357</td>
<td>2.214</td>
</tr>
<tr>
<td></td>
<td>2.526</td>
<td>2.368</td>
</tr>
<tr>
<td>Prosody</td>
<td>2.190</td>
<td>2.857</td>
</tr>
<tr>
<td></td>
<td>2.357</td>
<td>2.786</td>
</tr>
<tr>
<td></td>
<td>2.184</td>
<td>2.763</td>
</tr>
</tbody>
</table>

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 (∼0.351) is smaller than the average rating difference for the training stage (∼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.
### Average rating of LC

<table>
<thead>
<tr>
<th></th>
<th>Anno1</th>
<th>Anno2</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spelling</strong></td>
<td>1.288</td>
<td>1.318</td>
<td>0.030</td>
</tr>
<tr>
<td><strong>Inseparability</strong></td>
<td>2.303</td>
<td>2.591</td>
<td>0.288</td>
</tr>
<tr>
<td><strong>Inability to modify the modifier</strong></td>
<td>2.470</td>
<td>2.682</td>
<td>0.212</td>
</tr>
<tr>
<td><strong>Inability to replace the head by one</strong></td>
<td>1.909</td>
<td>2.364</td>
<td>0.455</td>
</tr>
<tr>
<td><strong>Inflection of the modifier</strong></td>
<td>1.909</td>
<td>2.364</td>
<td>0.455</td>
</tr>
<tr>
<td><strong>Prosody</strong></td>
<td>1.909</td>
<td>2.576</td>
<td>0.667</td>
</tr>
</tbody>
</table>

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.
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%.

<table>
<thead>
<tr>
<th>No.</th>
<th>LC Selection</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spelling</td>
<td>45.1%</td>
</tr>
<tr>
<td>2</td>
<td>Inseparability</td>
<td>50.7%</td>
</tr>
<tr>
<td>3</td>
<td>Inability to Modify the Modifier</td>
<td>50.1%</td>
</tr>
<tr>
<td>4</td>
<td>Replacement by ‘one’</td>
<td>48.2%</td>
</tr>
<tr>
<td>5</td>
<td>Inflection of the Modifier</td>
<td>48.2%</td>
</tr>
<tr>
<td>6</td>
<td>Prosody</td>
<td>48.9%</td>
</tr>
<tr>
<td></td>
<td>all LCs</td>
<td>63.7%</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Group ID</th>
<th>LC Numbers</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1,2,3,4,5,6</td>
<td>63.7%</td>
</tr>
<tr>
<td>B</td>
<td>2,3,4,5,6</td>
<td>63.1%</td>
</tr>
<tr>
<td>C</td>
<td>2,3,6</td>
<td>62.5%</td>
</tr>
<tr>
<td>D</td>
<td>2,3</td>
<td>56.4%</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>50.7%</td>
</tr>
<tr>
<td>F</td>
<td>1,4,5</td>
<td>47.2%</td>
</tr>
</tbody>
</table>

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 $p < 0.05$.

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>

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.
10. Pilot Studies on Compound Identification

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

<table>
<thead>
<tr>
<th>No.</th>
<th>LC Selection</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spelling</td>
<td>43.8%</td>
</tr>
<tr>
<td>2</td>
<td>Inseparability</td>
<td>48.9%</td>
</tr>
<tr>
<td>3</td>
<td>Inability to Modify the Modifier</td>
<td>50.1%</td>
</tr>
<tr>
<td>4</td>
<td>Replacement by ‘one’</td>
<td>48.2%</td>
</tr>
<tr>
<td>5</td>
<td>Inflection of the Modifier</td>
<td>48.2%</td>
</tr>
<tr>
<td>6</td>
<td>Prosody</td>
<td>48.9%</td>
</tr>
</tbody>
</table>

Table 10.13.: Compoundhood prediction accuracy using a Naive Bayes classifier

<table>
<thead>
<tr>
<th>Group ID</th>
<th>LC Numbers</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1,2,3,4,5,6</td>
<td>55.7%</td>
</tr>
<tr>
<td>B</td>
<td>2,3,4,5,6</td>
<td>57.4%</td>
</tr>
<tr>
<td>C</td>
<td>2,3,6</td>
<td>55.6%</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>50.1%</td>
</tr>
<tr>
<td>E</td>
<td>1,4,5</td>
<td>44.1%</td>
</tr>
</tbody>
</table>

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 $p < 0.05$.

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.

<table>
<thead>
<tr>
<th>Compoundhood</th>
<th>LC</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spelling</td>
<td>2.137</td>
<td>0.834</td>
</tr>
<tr>
<td>Inseparability</td>
<td>2.123</td>
<td>0.014</td>
</tr>
<tr>
<td>Inability to modify the modifier</td>
<td>2.244</td>
<td>0.107</td>
</tr>
<tr>
<td>Inability to replace the head by one</td>
<td>2.000</td>
<td>0.137</td>
</tr>
<tr>
<td>Inflection of the modifier</td>
<td>2.000</td>
<td>0.137</td>
</tr>
<tr>
<td>Prosody</td>
<td>2.084</td>
<td>0.053</td>
</tr>
</tbody>
</table>

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 $N_1 N_2$ in the English word sequence $ADJ N_1 N_2$, we could use the French phrasal equivalent $\tau(N_2) \text{ de } \tau(N_1) \tau(ADJ)$, where $\tau(\psi)$ refers to the French equivalent of $\psi$. If there is a morphological disagreement between $\tau(\text{adj})$ and $\tau(N_2)$ (e.g., in gender) but an agreement between $\tau(\text{adj})$ 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.
10. Pilot Studies on Compound Identification

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.
Firstly, all aligned languages are considered. The most frequent USPs (with a percentage above 1%) are given in Table 10.17.

<table>
<thead>
<tr>
<th>USP</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>24,487</td>
<td>29.7%</td>
</tr>
<tr>
<td>SN FC SN</td>
<td>22,173</td>
<td>26.9%</td>
</tr>
<tr>
<td>SN</td>
<td>9357</td>
<td>11.4%</td>
</tr>
<tr>
<td>SN SN</td>
<td>9347</td>
<td>11.3%</td>
</tr>
<tr>
<td>SN ADJ</td>
<td>8491</td>
<td>10.3%</td>
</tr>
<tr>
<td>ADJ SN</td>
<td>3765</td>
<td>4.6%</td>
</tr>
</tbody>
</table>

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%).

<table>
<thead>
<tr>
<th>USP</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>24,487</td>
<td>70.6%</td>
</tr>
<tr>
<td>SN</td>
<td>5735</td>
<td>16.5%</td>
</tr>
<tr>
<td>SN SN</td>
<td>1525</td>
<td>4.4%</td>
</tr>
<tr>
<td>ADJ SN</td>
<td>1129</td>
<td>3.3%</td>
</tr>
<tr>
<td>ADJ CN</td>
<td>559</td>
<td>1.6%</td>
</tr>
</tbody>
</table>

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.
Thirdly, we have a look at the Hellenic language family, i.e., on Greek. Table 10.19 shows the most frequent Greek USPs (with a percentage above 1%).

<table>
<thead>
<tr>
<th>USP</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN SN</td>
<td>5533</td>
<td>53.7%</td>
</tr>
<tr>
<td>ADJ SN</td>
<td>2418</td>
<td>23.5%</td>
</tr>
<tr>
<td>SN</td>
<td>887</td>
<td>8.6%</td>
</tr>
<tr>
<td>SN FC SN</td>
<td>599</td>
<td>5.8%</td>
</tr>
<tr>
<td>FC SN SN</td>
<td>136</td>
<td>1.3%</td>
</tr>
</tbody>
</table>

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 family, 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%).

<table>
<thead>
<tr>
<th>USP</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN FC SN</td>
<td>21,420</td>
<td>57.3%</td>
</tr>
<tr>
<td>SN ADJ</td>
<td>8450</td>
<td>22.6%</td>
</tr>
<tr>
<td>SN</td>
<td>2735</td>
<td>7.3%</td>
</tr>
<tr>
<td>SN SN</td>
<td>2289</td>
<td>6.1%</td>
</tr>
</tbody>
</table>

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
10. Pilot Studies on Compound Identification

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...
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 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 \texttt{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 \textit{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\(^1\) (last two rows of Table 11.1). As PoS tag set, we use the Penn Treebank tag set (Marcus et al., 1993).

<table>
<thead>
<tr>
<th>PoS pattern</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binary nominal compounds</strong></td>
<td></td>
</tr>
<tr>
<td>\texttt{NN}</td>
<td>\textit{marketplace}</td>
</tr>
<tr>
<td>\texttt{NN NN}</td>
<td>\textit{death penalty}</td>
</tr>
<tr>
<td>\texttt{JJ NN}</td>
<td>\textit{structural policy}</td>
</tr>
<tr>
<td>\texttt{NN POS NN}</td>
<td>\textit{children’s development}</td>
</tr>
<tr>
<td><strong>Ternary nominal compounds</strong></td>
<td></td>
</tr>
<tr>
<td>\texttt{NN NN NN}</td>
<td>\textit{energy security goal}</td>
</tr>
<tr>
<td>\texttt{JJ NN NN}</td>
<td>\textit{overall recovery rate}</td>
</tr>
<tr>
<td><strong>Regular expressions for</strong></td>
<td></td>
</tr>
<tr>
<td>\texttt{k-ary nominal compounds} ((2 \leq k \leq 10))</td>
<td></td>
</tr>
<tr>
<td>\texttt{NN (POS? NN)}{0,9}</td>
<td></td>
</tr>
<tr>
<td>\texttt{JJ NN (POS? JJ? NN)}{0,8}</td>
<td></td>
</tr>
</tbody>
</table>

Table 11.1.: Predefined PoS patterns for the compound candidate selection

For single nouns (e.g., \textit{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.

\(^1\)While the largest plausible word sequence observed in Europarl comprised 7 words, we defined an upper bound of 10 words.
11. Compound Identification Method

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\(^2\).

(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.

\[
P(\text{noun/adj} \mid \text{word}) = \frac{f((\text{noun} \cup \text{adj}) \cap \text{word})}{f(\text{word})} \tag{11.1}
\]

We disqualify English word sequences, if they contain a noun or adjective \(w\) with \(P(\text{noun/adj} \mid w) < \theta\). 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.

\(^2\text{ranks.nl/stopwords}\)
(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\(^3\), 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$^1$ rights$^2$ violations$^3$, we observed the following aligned sentence fragment in Italian: ...che le violazioni$^3$ gravi e sistematiche dei diritti$^2$ umani$^1$ ... ‘...that serious and systematic violations$^3$ of human$^1$ rights$^2$ ...’). 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 $AWS(rights)$ and $AWS(violations)$ 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.

\(^3\)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. Compound Identification Method

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:

\[ \text{CCR}(n): \text{An English compound candidate is considered to be a nominal compound if it is represented as a single noun in at least } n \text{ languages among the closed compounding support languages.} \]

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 (\( \Xi_{\text{closed}} \)) 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., \( \Xi_{\text{closed}} = 1 \)) might not be restrictive enough. The CCR with \( \Xi_{\text{closed}} \geq i \) includes only English compound candidates that are aligned to at least \( i \) single nouns in the aligned closed compounding languages (i.e., \( \text{CCR}(i) \)). We expect that \( \Xi_{\text{closed}} = j \) 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 \( \Xi_{\text{closed}} \), we compiled four versions of the database (i.e., \( \text{CCR}(1) \) to \( \text{CCR}(4) \)). As baseline, we used all compound candidates (i.e., \( \text{CCR}(0) \)). We expect \( \text{CCR}(0) \) to have the highest recall for compound identification, whereas \( \text{CCR}(4) \) 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 $\Xi_{\text{closed}}$, 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 $\Xi_{\text{closed}}$: 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 $\Xi_{\text{closed}}$ (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.
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).

Another trend we observed in Table 12.1 is that with increasing Ξclosed, 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

<table>
<thead>
<tr>
<th>CCR(0)</th>
<th>CCR(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>same time</td>
<td>agricultural policy</td>
</tr>
<tr>
<td>internal market</td>
<td>cohesion policy</td>
</tr>
<tr>
<td>next item</td>
<td>euro area</td>
</tr>
<tr>
<td>other hand</td>
<td>fisheries policy</td>
</tr>
<tr>
<td>great deal</td>
<td>security policy</td>
</tr>
<tr>
<td>European level</td>
<td>decision-making process</td>
</tr>
<tr>
<td>agricultural policy</td>
<td>development cooperation</td>
</tr>
<tr>
<td>last year</td>
<td>labour market</td>
</tr>
<tr>
<td>economic crisis</td>
<td>nuclear energy</td>
</tr>
<tr>
<td>single market</td>
<td>own-initiative report</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>CCR(0)</th>
<th>CCR(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>today’s debate</td>
<td>lion’s share</td>
</tr>
<tr>
<td>today’s vote</td>
<td>snail’s pace</td>
</tr>
<tr>
<td>tomorrow’s vote</td>
<td>women’s movement</td>
</tr>
<tr>
<td>rapporteur’s proposal</td>
<td>prosecutor’s office</td>
</tr>
<tr>
<td>world’s population</td>
<td>women’s issue</td>
</tr>
<tr>
<td>year’s budget</td>
<td>citizen’s initiative</td>
</tr>
<tr>
<td>minute’s silence</td>
<td>auditor’s report</td>
</tr>
<tr>
<td>rapporteur’s view</td>
<td>world’s history</td>
</tr>
<tr>
<td>today’s world</td>
<td>world’s economy</td>
</tr>
<tr>
<td>Today’s vote</td>
<td>women’s quota</td>
</tr>
</tbody>
</table>

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 $\Xi_{closed}$.  

155
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.

<table>
<thead>
<tr>
<th>Language</th>
<th>CCR(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danish</td>
<td>421K (60.6%)</td>
</tr>
<tr>
<td>Dutch</td>
<td>376K (54.1%)</td>
</tr>
<tr>
<td>German</td>
<td>451K (64.8%)</td>
</tr>
<tr>
<td>Swedish</td>
<td>335K (48.1%)</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>German</th>
<th>Swedish</th>
<th>French</th>
<th>Italian</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN CN</td>
<td>CN CN</td>
<td>SN FC SN FC SN</td>
<td>SN FC SN FC SN</td>
</tr>
<tr>
<td>ADJ CN</td>
<td>ADJ CN</td>
<td>SN FC SN ADJ</td>
<td>SN FC SN ADJ</td>
</tr>
<tr>
<td>SN FC CN</td>
<td>SN</td>
<td>SN FC SN</td>
<td>SN ADJ</td>
</tr>
<tr>
<td>SN SN FC SN</td>
<td>SN ADJ SN</td>
<td>SN SN ADJ</td>
<td>SN SN SN</td>
</tr>
<tr>
<td>ADJ SN</td>
<td>SN SN</td>
<td>SN FC SN</td>
<td>SN FC SN</td>
</tr>
<tr>
<td>SN SN SN</td>
<td>SN SN</td>
<td>SN FC ADJ SN</td>
<td>SN SN SN</td>
</tr>
<tr>
<td>SN FC ADJ SN</td>
<td>PC CN</td>
<td>SN FC SN SN</td>
<td>SN SN SN</td>
</tr>
<tr>
<td>SN SN SN</td>
<td>PC SN</td>
<td>SN FC SN</td>
<td>SN FC SN SN</td>
</tr>
<tr>
<td>CN FC SN</td>
<td>SN SN SN</td>
<td>SN</td>
<td>SN</td>
</tr>
<tr>
<td>CN FC CN</td>
<td>SN VB SN</td>
<td>SN SN ADJ</td>
<td>SN SN ADJ</td>
</tr>
</tbody>
</table>

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 Energieübertragungsvorluste 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
12. Europarl Nominal Compound Database

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, $\Xi_{\text{closed}}$, 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
13. Experiments on Compound Identification

- resulting Jaccard coefficient
- Precision, Recall and F\textsubscript{1}-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\textsubscript{1}-Score of 0.603 observed for the IAA between the two annotators, described in Table 10.3.

<table>
<thead>
<tr>
<th>System</th>
<th># extractions</th>
<th>Jaccard</th>
<th>P</th>
<th>R</th>
<th>F\textsubscript{1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCR(0)</td>
<td>355</td>
<td>0.262</td>
<td>0.327</td>
<td>0.569</td>
<td>0.415</td>
</tr>
<tr>
<td>CCR(1)</td>
<td>174</td>
<td>0.245</td>
<td>0.669</td>
<td>0.279</td>
<td>0.394</td>
</tr>
<tr>
<td>CCR(2)</td>
<td>124</td>
<td>0.189</td>
<td>0.790</td>
<td>0.199</td>
<td>0.318</td>
</tr>
<tr>
<td>CCR(3)</td>
<td>82</td>
<td>0.128</td>
<td>0.837</td>
<td>0.131</td>
<td>0.227</td>
</tr>
<tr>
<td>CCR(4)</td>
<td>41</td>
<td>0.065</td>
<td>0.932</td>
<td>0.066</td>
<td>0.123</td>
</tr>
</tbody>
</table>

Table 13.1.: Compound Identification Results for \textit{Annotation1} dataset

Table 13.1 shows the results for the \textit{Annotation1} dataset, including annotations of the first annotator only. The best identification quality in terms of Jaccard coefficient and F\textsubscript{1}-Score is achieved for the ENCD version with \( \Xi_{\text{closed}} \geq 0 \), 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 \( \Xi_{\text{closed}} \), outweighing the precision gain.

Table 13.2 shows the results for the \textit{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 \( \Xi_{\text{closed}} \). As a consequence, the best identification quality is achieved for CCR(1) with a Jaccard coefficient of 0.196 and an F\textsubscript{1}-Score of 0.318.
13. Experiments on Compound Identification

### Table 13.2: Compound Identification Results for Annotation2 dataset

<table>
<thead>
<tr>
<th>System</th>
<th># extractions</th>
<th>Jaccard</th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCR(0)</td>
<td>151 742</td>
<td>0.161</td>
<td>0.204</td>
<td>0.436</td>
<td>0.278</td>
</tr>
<tr>
<td>CCR(1)</td>
<td>346</td>
<td>0.196</td>
<td>0.470</td>
<td>0.251</td>
<td>0.328</td>
</tr>
<tr>
<td>CCR(2)</td>
<td>87 185</td>
<td>0.148</td>
<td>0.527</td>
<td>0.171</td>
<td>0.258</td>
</tr>
<tr>
<td>CCR(3)</td>
<td>59 112</td>
<td>0.111</td>
<td>0.568</td>
<td>0.121</td>
<td>0.200</td>
</tr>
<tr>
<td>CCR(4)</td>
<td>42 74</td>
<td>0.048</td>
<td>0.630</td>
<td>0.049</td>
<td>0.091</td>
</tr>
</tbody>
</table>

Table 13.2 shows the results for the Annotation2 dataset, including annotations of both annotators. The best identification quality is achieved for CCR(1) (Jaccard = 0.236, F₁-Score = 0.382).

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.

### Table 13.3: Compound Identification Results for Combination dataset

<table>
<thead>
<tr>
<th>System</th>
<th># extractions</th>
<th>Jaccard</th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCR(0)</td>
<td>450 1555</td>
<td>0.233</td>
<td>0.289</td>
<td>0.546</td>
<td>0.378</td>
</tr>
<tr>
<td>CCR(1)</td>
<td>230 381</td>
<td>0.236</td>
<td>0.604</td>
<td>0.279</td>
<td>0.382</td>
</tr>
<tr>
<td>CCR(2)</td>
<td>163 237</td>
<td>0.182</td>
<td>0.688</td>
<td>0.198</td>
<td>0.307</td>
</tr>
<tr>
<td>CCR(3)</td>
<td>111 151</td>
<td>0.128</td>
<td>0.735</td>
<td>0.135</td>
<td>0.228</td>
</tr>
<tr>
<td>CCR(4)</td>
<td>52 62</td>
<td>0.062</td>
<td>0.839</td>
<td>0.063</td>
<td>0.117</td>
</tr>
</tbody>
</table>

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, F₁-Score = 0.382).

In general, all three tables show a strong recall drop and precision gain when increasing Ξclosed. 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 F₁-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 focusing 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
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 F1-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 ($\Xi_{\text{closed}}$) 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 $\Xi_{\text{closed}} = 4$.

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 $\Xi_{\text{closed}}$. 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 $\oplus$s-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: *Eidot*|*tter* ‘oath otter’.

![Figure 15.1.: Structures for the German Hochschulgebäude ‘university building’](image)

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.
15. Introduction to Compound Splitting

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 Kranken|versicherung ‘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., $Ei\dot{ot}ter$ `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?
15. Introduction to Compound Splitting

• 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?
15. Introduction to Compound Splitting

- What methods perform best with respect to split point determination?
- What systems are superior in constituent normalization?

**RQ_2-E-i:** 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
16. Related Work on Compound Splitting

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\(^1\) 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 produce 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.

\(^1\)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, \( F_1 \)-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 tasks 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.

<table>
<thead>
<tr>
<th>Target word</th>
<th>F</th>
<th>r</th>
<th>i</th>
<th>e</th>
<th>d</th>
<th>e</th>
<th>n</th>
<th>s</th>
<th>p</th>
<th>o</th>
<th>l</th>
<th>i</th>
<th>t</th>
<th>i</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>counts</td>
<td>prefix</td>
<td>-</td>
<td>-</td>
<td>39</td>
<td>29</td>
<td>29</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>suffix</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>37</td>
<td>88</td>
<td>89</td>
<td>92</td>
<td>99</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Δ counts</td>
<td>prefix</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>suffix</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>30</td>
<td>51</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>max Δ counts</td>
<td>prefix</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>suffix</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 *Ei*|*dotter* ‘oath otter’. While *Ei*|*dotter* has the phonetic transcription *[ai]|*|*dt5*], *Eid|otter* is pronounced as *[ait]|*|*t5*] (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 +s suffix, whereas Koehn and Knight (2003), as will be discussed shortly, used two fillers for normalization: +s and +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.

<table>
<thead>
<tr>
<th>Constituents</th>
<th>Frequencies</th>
<th>Geometric mean score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aktionsplan ‘action plan’</td>
<td>852</td>
<td>852</td>
</tr>
<tr>
<td>Aktion ‘action’ • Plan ‘plan’</td>
<td>960 • 710</td>
<td>825.6</td>
</tr>
<tr>
<td>Aktions ‘action (s-suffix)’ • Plan ‘plan’</td>
<td>5 • 710</td>
<td>59.6</td>
</tr>
<tr>
<td>Akt ‘act’ • Ion ‘ion’ • Plan ‘plan’</td>
<td>224 • 1 • 710</td>
<td>54.2</td>
</tr>
</tbody>
</table>

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 constituents and compounds
- Maximum number of constituents
- Valid compound class (content words vs. all words)
- Optional PoS equality between head and compound
- Additional morphological operations (e.g., those proposed by Langer (1998))
- Corpus frequencies of constituent forms and/or constituent lemmas
- Hyphens as exclusive split points for hyphenated compounds

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., stopplikt → stopp plikt ‘stop obligation’). Besides the operations proposed by Langer (1998), Stymne and Holmqvist (2008) used two additive suffixes $+$s and $+$t, two subtractive suffixes $-$e and $-$a, as well as nine combinations of suffixation and truncation (e.g., $\ominus$a/$\ominus$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.


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: $+$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
16. Related Work on Compound Splitting

with string similarity (e.g., the normalized Levenshtein distance (Frunza and Inkpen, 2009)) to lemmas within a monolingual lexicon including PoS-filtered lemmas and neoclassical 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., +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.

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\(^2\) 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

\(^2\)Brown (2002) used the least common substring (LCS) as string similarity measure.
16. Related Work on Compound Splitting

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*- ‘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. The wFST is automatically created using the AT&T FSM Library 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

---

3This dataset will be used as a compound splitting gold standard in Chapter 18.
4https://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|auf|nahme ‘tape recording’. For a translation to English, all paths except for splitting aufnahme are possible.

![Split lattice example](image)

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 Ngrams 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 \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. Related Work on Compound Splitting

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 dependend 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., *losfahren* ‘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\(^5\). 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

\(^5\) [http://www.xrce.xerox.com/competencies/content-analysis/demos/german](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 Arbeitnehmer ‘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: +nen, +ien, +en, +er and +n, as well as two truncatable suffixes: +e and +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-
16. Related Work on Compound Splitting

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. A fourth linguistically motivated method, mentioned by Escartín (2014) as part of future work, is the smor-based system of Fritzinger 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU</th>
<th>TER</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>45.9%</td>
<td>43.9%</td>
<td>181</td>
</tr>
<tr>
<td>Popović et al. (2006)</td>
<td>48.3%</td>
<td>40.8%</td>
<td>104</td>
</tr>
<tr>
<td>Weller and Heid (2012)</td>
<td>48.3%</td>
<td>40.5%</td>
<td>114</td>
</tr>
</tbody>
</table>

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

---

6http://www.danielnaber.de/jwordsplitter/index_en.html
16. Related Work on Compound Splitting

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASELINE</td>
<td>–</td>
<td>0%</td>
<td>89.79%</td>
</tr>
<tr>
<td>Popović et al. (2006) (TRIS+EUROPARL)</td>
<td>96.12%</td>
<td>72.51%</td>
<td>97.19%</td>
</tr>
<tr>
<td>Weller and Heid (2012)</td>
<td>99.23%</td>
<td>75.73%</td>
<td>97.49%</td>
</tr>
</tbody>
</table>

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
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 $\text{Hühner|suppe}$, the modifier lemma $\text{Huhn}$ is transformed to $\text{Hühner}$ by replacing $u$ by $ü$ (i.e., Umlautung) and adding the suffix $\text{er}$. The corresponding MOP is ‘$u/ü:\$/\text{er}$’. The second column in Table 17.1 shows some examples for the Germanic languages: German, Dutch and Afrikaans.
**17. Morphological Operation Patterns**

<table>
<thead>
<tr>
<th>Language</th>
<th>MOP</th>
<th>Frequency</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>u/ü$/er$</td>
<td>291</td>
<td>&lt;Huhn, Hühner&gt; ‘chicken’,&lt;Buch, Bücher&gt; ‘book’</td>
</tr>
<tr>
<td></td>
<td>um$/en$</td>
<td>629</td>
<td>&lt;Studium, Studien&gt; ‘study’,&lt;Medium, Medien&gt; ‘medium’</td>
</tr>
<tr>
<td>Dutch</td>
<td>$/en$</td>
<td>7050</td>
<td>&lt;arts, artsen&gt; ‘doctor’,&lt;band, banden&gt; ‘tyre’</td>
</tr>
<tr>
<td>Afrikaans</td>
<td>$/se$</td>
<td>2768</td>
<td>&lt;proses, prosesse&gt; ‘process’</td>
</tr>
</tbody>
</table>

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 Σ and Ω. 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 $α_{form}|β_{form}$ analyzed into the lemmas $α_{lem}$ and $β_{lem}$, the corresponding MOPs are derived from the string pairs $(α_{lem},α_{form})$ and $(β_{lem},β_{form})$, i.e., MOP[$α_{lem}→α_{form}$] and MOP[$β_{lem}→β_{form}$]. Subsequently, MOPs derived from gold compound splits are called **gold-constituent MOPs**.

196
17.2.3. Hand-crafted constituent MOPs

The transparent syntax of MOPs allows for a manual typing of the MOPs\(^1\). 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, MOP\(_{$/er$/}(Kind) = Kinder

- The substring replacements \(\mu\) are applied sequentially starting with the first.
  For example, MOP\(_{u/ü:$/er$/}(Huhn) = 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, MOP\(_{u/ü:$/er$/}(Kind) = 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

\(^1\)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}_u /ü:$/er$ (Suppenhuhn) = Suppenhühner ‘boiling hen’ or \( \text{MOP}_u /ä:$/e$ (Patentanwalt) = Patentanwälte ‘patent attorney’ \)

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}[\text{Huhn} \rightarrow \text{Hühner}] = u/ü:$/er$. 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}[\text{Huhn} \rightarrow \text{Hühner}] = u/ü:$/er$. Inverting this MOP results in: \( \text{MOP}[\text{Hühner} \rightarrow \text{Huhn}] = ü/u:er$/$. Re-applying the inverted MOP to Hühner results in the initial lemma: \( \text{MOP}_u/ü:er$/ (\text{Hühner}) = \text{Huhn} \).
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\(^1\). The constituents’ word category have to be any content word type. Lemmas which are function words are excluded as constituents\(^2\). 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).

\(^1\)The hierarchical and binary splitting architecture requires that the immediate constituents at each level must be able to be mapped on known corpus lexemes.

\(^2\)This prevents oversplits into functional head lemmas, e.g., Bohrer ‘driller’ would be split into bohr | er ‘drilling he’.
18. Multilingual Compound Splitting

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, . . . , Ö|lpre|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

Figure 18.1.: Architecture of the splitting algorithm
18. Multilingual Compound Splitting

TV-Programm ‘TV program’. Alternative split point markers\(^3\) 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 Ngram index for constituent normalization. In this Lemma Resource (LR), Ngrams 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 Ngrams with a

---

\(^3\)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. \( \text{CLs} \leftarrow \{ \} \) \{the candidate lemma (mapping lemmas to the score)\}
2. for \( MOP_i \) in MR do
3. \( IM_i \leftarrow \text{invert}(MOP_i) \)
4. \( l_i \leftarrow IM_i(c) \)
5. if \( l_i \neq \text{NULL} \) then
6. \( \text{CLs}[l_i] = \text{score}(l_i, \text{LR}, \text{MOP}_i, \text{MR}) \)
7. end if
8. end for
9. rank(\text{CLs})
10. return top\( M \) (\text{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 \text{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 \( \text{CLs} \) (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.

\[
MS(l_i) = \frac{\log_{10}(freq(MOP[l_i \rightarrow c]))}{ED[l_i \rightarrow c] + 1}
\]  

The MS estimates the suitability of using the MOP transforming \( l_i \) for the constituent form at hand, \( c \), (represented as \( MOP[l_i \rightarrow c] \)) as a candidate for constituent inflection. As the first component, we use the \( \log_{10} \) of the frequency associated with the MOP, \( \log_{10}(freq(MOP[l_i \rightarrow c])) \). This value is rescaled with the resulting \( ED \) between the candidate lemma \( l_i \) and the constituent form, \( c \), (represented as \( ED[l_i \rightarrow c] \)). 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 \( MOP[\text{wees} \rightarrow \text{is}] = \text{\^{\text{\text{\^}}}w\text{\text{\text{e}}}}/\text{\text{\text{\text{\text{\text{\text{i}}}}}}}} \).
18. Multilingual Compound Splitting

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).

\[
\text{score}(l_i) = LF(l_i) \cdot MS(l_i)
\]  

(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\(^4\) 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, \(con = \{l_h\}\).

\[
\text{geo}(con) = \sqrt[|con|]{\prod_{l_i \in con} \text{score}(l_i)}
\]  

(18.3)

The second feature of the combination model is motivated by the RightHand Head Rule (RHHHR), 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 \(p(\text{PoS}|\text{word}) = \frac{\text{freq}(\text{PoS, word})}{\text{freq}(\text{word})}\) acquired from the monolingual PoS-tagged corpus. The value of the head-PoS-EQuality (hEQ) feature is defined as the 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.

\[
\text{split}(con) = \text{geo}(con) \cdot hEQ(\Psi, l_h)
\]  

(18.4)

\(^4\)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\(^5\) 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

\(^5\)In Ziering and Van der Plas (2016), we did not use these features.
18. Multilingual Compound Splitting

constituent), there is a high risk of using a misleading word MOP that yields a false binary split. For example, *Termine* 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.

<table>
<thead>
<tr>
<th>Word MOP</th>
<th>Word form</th>
<th>Number</th>
<th>Case</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>/</em> (null-MOP)</td>
<td><em>Termin</em></td>
<td>singular</td>
<td>nominative</td>
</tr>
<tr>
<td>$/$es$</td>
<td><em>Termins</em></td>
<td></td>
<td>accusative</td>
</tr>
<tr>
<td>$$/es$$</td>
<td><em>Termines</em></td>
<td></td>
<td>dative</td>
</tr>
<tr>
<td>$$/e$$</td>
<td><em>Termine</em></td>
<td>plural</td>
<td>genitive</td>
</tr>
<tr>
<td>$$/en$$</td>
<td><em>Terminen</em></td>
<td></td>
<td>nominative</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>accusative</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>genitive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dative</td>
</tr>
</tbody>
</table>

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 ($\text{MOP}^{\text{inv}}_\alpha$) 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, *Termine* is normalized to *Termin* by applying the inversed MOP $e$s/$,$, 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|hald ‘as soon as’, 
dal|nach ‘afterwards’, dal|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)’](image)

Using MOP application with a PoS agreement (similar to the lexeme agreement de-
scribed in Section 18.4.1), it is possible to reduce erroneous splittings which are due to 
cross-categorial MOP application.

Therefor, 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 (MOP\(_a\)), the inversion (MOP\(_{\text{inv}}\)) may only be applied 
to a constituent form if there is at least one common PoS between the collection 
for the original word MOP (MOP\(_a\)) 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.

<table>
<thead>
<tr>
<th>Word MOP</th>
<th>Word form</th>
<th>Number</th>
<th>Case</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>/</em> (null-MOP)</td>
<td>Lauf</td>
<td>singular</td>
<td>nominative</td>
</tr>
<tr>
<td>$$/s$$</td>
<td>Laufs</td>
<td></td>
<td>genitive</td>
</tr>
<tr>
<td>$$/es$$</td>
<td>Laufes</td>
<td></td>
<td>genitive</td>
</tr>
<tr>
<td>$$/e$$</td>
<td>Laufe</td>
<td></td>
<td>dative</td>
</tr>
<tr>
<td>a/a:$/e$</td>
<td>Läufe</td>
<td>plural</td>
<td>nominative</td>
</tr>
<tr>
<td>a/ä:$/en$</td>
<td>Läufen</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 18.2.: Word MOPs for the lemma **Lauf** ‘run’

The frequent word MOP a/ä:$/er$ (as in <Mann,Männer> ‘<man,mens>’) 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/ä:$er$ (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.

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 Kindheit|erinnerung ‘childhood memory’), Kindheits is not a paradigmatic word form and thus the word MOP $/$ 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](image-url)

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., Fußballländerspiel ‘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 $1 \leq k_i \leq k_{\text{init}}$, where $k_{\text{init}}$ 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.

First pruning step, $k_i = 3$

Second pruning step, $k_i = 2$

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
18. Multilingual Compound Splitting

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 LSF\textsubscript{word}. For example, the LSF\textsubscript{word} for the pluralized \textit{Hühnersuppenrezepte} ‘chicken soup recipes’ is \textit{Huhn Suppe Rezept}. A further motivation for using the LSF\textsubscript{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 LSF\textsubscript{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 \textit{Suppenhühnerzucht} ‘boiling hen breeding’ is represented in the SPF as: \textit{Suppen} | \textit{hühner} | \textit{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: \textit{German} (Section 3.9.2), \textit{Dutch} (Section 3.9.3) and \textit{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\(^6\).

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 TreeTagger\(^7\) (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\(^8\) 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

\(^6\)\{de.nl.af\}.wikipedia.org
\(^7\)www.cis.uni-muenchen.de/~schmid/tools/TreeTagger
\(^8\)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., üß$/üsse$). 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.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Language</th>
<th>Tokens</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>words</td>
<td>lemmas</td>
</tr>
<tr>
<td>WIKIPEDIA</td>
<td>German</td>
<td>667.3M</td>
<td>9.0M</td>
</tr>
<tr>
<td></td>
<td>Dutch</td>
<td>115.3M</td>
<td>2.0M</td>
</tr>
<tr>
<td></td>
<td>Afrikaans</td>
<td>12.0M</td>
<td>370.6K</td>
</tr>
</tbody>
</table>

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.
18. Multilingual Compound Splitting

<table>
<thead>
<tr>
<th>Target compound</th>
<th>SPF</th>
<th>LSF</th>
<th>Split tree bracketing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fußbodenheizung</td>
<td>Fuß</td>
<td>boden</td>
<td>heizung</td>
</tr>
<tr>
<td>Fußbodenheizung</td>
<td>Fuß</td>
<td>boden</td>
<td>heizung</td>
</tr>
<tr>
<td>Fußbodenheizung</td>
<td>Fußbodenheizung</td>
<td>Fußbodenheizung</td>
<td>Fußbodenheizung</td>
</tr>
</tbody>
</table>

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 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, 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 Fern sprechen | 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.

9sfs.uni-tuebingen.de/GermaNet
10http://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).

<table>
<thead>
<tr>
<th>word MOPs</th>
<th>word MOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOPs in HH2011GS</td>
<td>54</td>
</tr>
<tr>
<td>MOPs in HH2011GS</td>
<td>1141</td>
</tr>
</tbody>
</table>

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 (\(~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 *ießen$/uss$*, 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 *$/o$*. This MOP 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*\(^{11}\) 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, the 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.,

\(^{11}\)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., Vorstellung ‘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.

<table>
<thead>
<tr>
<th>Compound size</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>116,218 (83.6%)</td>
</tr>
<tr>
<td>3</td>
<td>21,651 (15.6%)</td>
</tr>
<tr>
<td>4</td>
<td>1175 (0.84%)</td>
</tr>
<tr>
<td>5</td>
<td>35 (0.03%)</td>
</tr>
<tr>
<td>6</td>
<td>2 (0.001%)</td>
</tr>
<tr>
<td>total</td>
<td>139,081</td>
</tr>
</tbody>
</table>

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\}+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 $\text{tanzen}$ ‘to dance’ or $\text{Tanz}$ ‘dance’), it is implausible for some other modifiers (e.g., the modifier $\text{Stift}$ ‘pen’ in the compound $\text{Stiftgröße}$ ‘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 $\text{M2006GS}$ 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., $\text{tanz}_-$ instead of $\text{tanzen}$). 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 $\text{en}$ (e.g., $\text{tanz}$ $\rightarrow$ $\text{tanzen}$) or (2) adding the suffix $\text{n}$ (e.g., $\text{jammer}$ $\rightarrow$ $\text{jammern}$ ‘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 $\text{en}$ suffix is used as default.

In an experiment with 100 verb type samples, it turned out that 99 verb stems have correctly been transformed to the corresponding verb lemma. The single erroneous transformation is based on no corpus evidence and the default $\text{en}$ suffix. 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) $\text{M2006GS}$ with the regular LSF, exclusively containing bare constituent lemmas, and (2) $\text{M2006GS}$ with the LSF$_{\text{word}}$, which keeps the word-inflected compound head, as discussed in Section 18.5.2. For example, for the pluralized target compound $\text{Umweltschutzgebieten}$ ‘environmental protection areas’, the corresponding LSFs are: $\text{Umwelt Schutz Gebiet}$ (LSF) and $\text{Umwelt Schutz Gebieten}$ (LSF$_{\text{word}}$).

Creation of the SPF and LSF The initial version of $\text{M2006GS}$ does not directly provide an SPF or LSF, but a morphological parse which includes most information for compiling

---

12 The $\text{en}$ suffix is used in 81% of all cases for transforming a verb stem to the verb lemma.
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\textsubscript{word}). 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.

<table>
<thead>
<tr>
<th></th>
<th>HH2011GS</th>
<th>M2006GS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HH2011GS</td>
<td>16,252</td>
<td>—</td>
</tr>
<tr>
<td>M2006GS</td>
<td>37,900</td>
<td>—</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Compound size</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>637 (91%)</td>
</tr>
<tr>
<td>3</td>
<td>50 (7.1%)</td>
</tr>
<tr>
<td>1</td>
<td>13 (1.9%)</td>
</tr>
<tr>
<td>total</td>
<td>700</td>
</tr>
</tbody>
</table>

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.
18. Multilingual Compound Splitting

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.

<table>
<thead>
<tr>
<th></th>
<th>HH2011GS</th>
<th>HH2011GS</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2006GS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HB2008GS</td>
<td>306</td>
<td>381</td>
</tr>
<tr>
<td>HH2011GS</td>
<td>176,675</td>
<td>—</td>
</tr>
</tbody>
</table>

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 \texttt{word} (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\textsuperscript{13} 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.

\textsuperscript{13}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.
### 18. Multilingual Compound Splitting

#### 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).

<table>
<thead>
<tr>
<th>MOPs in VZDH2014GS/NL</th>
<th>word MOPs</th>
<th>word MOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MOPs in VZDH2014GS/AF</th>
<th>word MOPs</th>
<th>word MOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>2 (0)</td>
<td></td>
</tr>
</tbody>
</table>

Table 18.11.: Overlap between German word MOPs and gold-constituent MOPs in VZDH2014GS/AF
18. Multilingual Compound Splitting

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 x

(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 x
   → Kredit|karten|an|bieter x
   → Kreditkarten|anbieter x

(e) wrong split (ws): words that should not be split, but were
   e.g., Fenster ‘window’ → Fen|ster x

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).

\[ P = \frac{cs}{cs + wf + ws} \] (18.5)
18. Multilingual Compound Splitting

\[ R = \frac{cs}{cs + wn + wf} \]  \hspace{1cm} (18.6)

\[ Acc = \frac{cs + cn}{cs + cn + wn + wf + ws} \]  \hspace{1cm} (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 \( (P \equiv Acc) \) and precision is always as high as recall or higher \( (P \geq R) \).

Additionally, as a combination of precision and recall, the harmonic mean (F1-Score) is used as shown in Formula 18.8.

\[ F_1 \text{-Score} = \frac{2 \cdot P \cdot R}{P + R} \]  \hspace{1cm} (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 \( (SP_X, X \in \{ P, R, Acc, F_1 \}) \)
2. Normalization of the resulting constituent forms \( (Norm_X, X \in \{ P, R, Acc, F_1 \}) \)

All four measurements (i.e., precision, recall, accuracy and F-score) are represented for both disciplines, \( SP_X \) and \( Norm_X \), 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.

Fritzinger and Fraser (2010)

**Word-inflected Heads** The SMOR- (Schmid et al., 2004) and corpus-based approach of Fritzinger 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$_{\text{word}}$. 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$_{\text{word}}$. While the gold standard HB2008GS contains word-inflected compounds, it only provides an SPF and thus, no LSF$_{\text{word}}$ 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 version has been used. It relies on a PoS$_{15}$-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.$^{16}$

---

$^{14}$Release date: 13th October 2016
$^{15}$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)
$^{16}$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\textsubscript{word}. 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 \textit{Abenteuerromane} ‘adventure novels’ and the compound \textit{Hühnerfutter} ‘chicken feed’.

<table>
<thead>
<tr>
<th>Target compound</th>
<th>PoS-tagged LSF</th>
<th>PoS-tagged LSF\textsubscript{word}</th>
</tr>
</thead>
<tbody>
<tr>
<td>abenteuerromane</td>
<td>abenteuer_NN roman_NN</td>
<td>abenteuer_NN romane_NN</td>
</tr>
<tr>
<td>abenteuerromane</td>
<td>abenteuerroman_NN</td>
<td>abenteuerroman_NN</td>
</tr>
<tr>
<td>abenteuerromane</td>
<td>abenteuern_V roman_NN</td>
<td>abenteuern_V romane_NN</td>
</tr>
<tr>
<td>abenteuerromane</td>
<td>abene_NN teuer_NN roman_NN</td>
<td>abene_NN teuer_NN romane_NN</td>
</tr>
<tr>
<td>abenteuerromane</td>
<td>aben_NN teuer_NN roman_NN</td>
<td>aben_NN teuer_NN romane_NN</td>
</tr>
<tr>
<td>abenteuerromane</td>
<td>abente_NN euer_NN roman_NN</td>
<td>abente_NN euer_NN romane_NN</td>
</tr>
<tr>
<td>hühnerfutter</td>
<td>luhn_NN futter_NN</td>
<td>luhn_NN futter_NN</td>
</tr>
<tr>
<td>hühnerfutter</td>
<td>hühnerfutter_NN</td>
<td>hühnerfutter_NN</td>
</tr>
</tbody>
</table>

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\textsubscript{word} as well as any PoS tags.

Creation of the Split Point Format  As illustrated for the output of \textit{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 \textit{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\(^{17}\) deterioration in performance when subtracting it from the DEFAULT setup for at least some metrics.

\(^{17}\)Approximate Randomization Test (Yeh, 2000), \(p < 0.05\)
## 18. Multilingual Compound Splitting

<table>
<thead>
<tr>
<th>Feature</th>
<th>SPX</th>
<th>NormX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>97.3</td>
<td>96.4</td>
</tr>
<tr>
<td>☑ Constituent content words</td>
<td>96.4</td>
<td>95.6</td>
</tr>
<tr>
<td>☑ hEQ</td>
<td>97.4</td>
<td>96.8</td>
</tr>
<tr>
<td>☑ Prior MOP lemmatization</td>
<td>97.0</td>
<td>96.3</td>
</tr>
<tr>
<td>☑ Compound content words</td>
<td>97.3</td>
<td>96.3</td>
</tr>
<tr>
<td>☑ Modifier MOP PoS agreement</td>
<td>97.2</td>
<td>96.4</td>
</tr>
<tr>
<td>☑ Head MOP lexeme agreement</td>
<td>97.3</td>
<td>96.6</td>
</tr>
<tr>
<td>☑ Arithmetic mean</td>
<td>88.9</td>
<td>88.5</td>
</tr>
<tr>
<td>☑ Harmonic mean</td>
<td>96.6</td>
<td>94.9</td>
</tr>
</tbody>
</table>

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 (⭐️)
Table 18.13.: Results for German compound splitting - MOP set comparison; numbers in %

<table>
<thead>
<tr>
<th>Gold Standard</th>
<th>MOP Set</th>
<th>SPX</th>
<th>NormX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$P$</td>
<td>$R$</td>
</tr>
<tr>
<td>HH2011GS</td>
<td>word</td>
<td>98.1</td>
<td>96.9</td>
</tr>
<tr>
<td></td>
<td>HH2011GS</td>
<td>98.0</td>
<td>96.8</td>
</tr>
<tr>
<td></td>
<td>Langer</td>
<td>98.4</td>
<td>96.9</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>95.2</td>
<td>77.3</td>
</tr>
<tr>
<td>M2006GS</td>
<td>word</td>
<td>97.3</td>
<td>96.4</td>
</tr>
<tr>
<td></td>
<td>M2006GS</td>
<td>97.4</td>
<td>96.7</td>
</tr>
<tr>
<td></td>
<td>Langer</td>
<td>97.5</td>
<td>96.5</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>90.3</td>
<td>76.2</td>
</tr>
<tr>
<td>HB2008GS</td>
<td>word</td>
<td>98.7</td>
<td>97.1</td>
</tr>
<tr>
<td></td>
<td>Langer</td>
<td>98.7</td>
<td>96.2</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>94.1</td>
<td>74.4</td>
</tr>
</tbody>
</table>

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 $\text{eh}/\text{ang}$, which is usually used for transforming prefix verbs with the verb stem gehen ‘to go’ into the participle form, e.g., umgehen $\rightarrow$ 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 $\text{e/i}:\text{en}$/ for the imperative form of some German verbs, e.g., versprechen $\rightarrow$ versprich ‘to promise’). Since this operation of word inflection is never observed (for modifiers) in a compound splitting gold standard, this false normalization...
18. Multilingual Compound Splitting

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 \( Fördermittel \) ‘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 \( o/ö$/\$er$/ \), 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\(^{18}\).

Gold and Hand-crafted Constituent MOPs  The approach of gold-constituent MOPs, which can be considered as an \textit{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 \( Textiltechniker \) ‘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 \( Spiegelbilder \) ‘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 (\( \varphi \)) for all three gold standards. However, precision for both split point determination (\( \text{SPP} \)) and constituent normalization (\( \text{NormP} \)) 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 \( \text{HH2011GS} \) for the gold standards also providing compounds with three or more constituents, the

\(^{18}\)This holds for a language’s PoS tag set in which named entities have a unique PoS.
18. Multilingual Compound Splitting

...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.

<table>
<thead>
<tr>
<th>Gold Standard</th>
<th>MOP Set</th>
<th>SPX</th>
<th>NormX</th>
</tr>
</thead>
</table>
|               |           | P   | R    | Acc  | F
| DUTCH         |           |     |      |      |     |
| VZDH2014GS/NL | word      | 96.2| 94.6 | 95.4 | 81.0 |
|                | VZDH2014GS/NL | 97.0| 91.8 | 94.3 | 93.0 |
|                | null      | 93.9| 81.9 | 87.5 | 74.3 |
|                |           |     |      |      |     |
| AFRICAANS     |           |     |      |      |     |
| VZDH2014GS/AF | word      | 93.1| 82.9 | 87.7 | 87.0 |
|                | VZDH2014GS/AF | 94.0| 82.1 | 87.6 | 90.8 |
|                | null      | 89.5| 63.4 | 74.3 | 82.2 |

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
Table 18.15.: Results for German compound splitting - external system comparison

<table>
<thead>
<tr>
<th>Gold Standard</th>
<th>System</th>
<th>SPX</th>
<th>NormX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>HH2011GS</td>
<td>ZvdP2016</td>
<td>98.1</td>
<td>96.9</td>
</tr>
<tr>
<td></td>
<td>FF2010</td>
<td>98.5</td>
<td>92.3</td>
</tr>
<tr>
<td></td>
<td>WH2012</td>
<td>98.1</td>
<td>96.8</td>
</tr>
<tr>
<td>M2006GS</td>
<td>ZvdP2016</td>
<td>97.3</td>
<td>96.4</td>
</tr>
<tr>
<td></td>
<td>FF2010</td>
<td>92.9</td>
<td>88.2</td>
</tr>
<tr>
<td></td>
<td>WH2012</td>
<td>98.9</td>
<td>98.6</td>
</tr>
<tr>
<td>HB2008GS</td>
<td>ZvdP2016</td>
<td>98.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FF2010</td>
<td>96.6</td>
<td>91.4</td>
</tr>
<tr>
<td></td>
<td>WH2012</td>
<td>98.4</td>
<td>98.4</td>
</tr>
</tbody>
</table>

**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 *Funknetzwerk|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.
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.

<table>
<thead>
<tr>
<th>k</th>
<th>Distribution for ZvdP2016</th>
<th>Distribution for FF2010</th>
<th>Distribution for WH2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>139,081</td>
<td>139,081</td>
<td>91,665</td>
</tr>
<tr>
<td>2</td>
<td>137,899</td>
<td>132,170</td>
<td>137,939</td>
</tr>
<tr>
<td>3</td>
<td>38,828</td>
<td>15,383</td>
<td>96,092</td>
</tr>
<tr>
<td>4</td>
<td>6961</td>
<td>538</td>
<td>32,141</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
<td>9</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>109</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

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*.
18. Multilingual Compound Splitting

<table>
<thead>
<tr>
<th>Target compound</th>
<th>Splitting of ZvdP2016</th>
<th>Splitting of WH2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gastgeber</td>
<td>netzes ‘host network’</td>
<td>Gastgeber</td>
</tr>
<tr>
<td>Leinwand</td>
<td>größe ‘canvas size’</td>
<td>Leinwand</td>
</tr>
<tr>
<td>Multimedia</td>
<td>esse ‘multimedia fair’</td>
<td>Multimedia</td>
</tr>
<tr>
<td>Fotowerkzeuge ‘photo tools’</td>
<td>Fotowerkzeuge</td>
<td>Fotowerkzeuge</td>
</tr>
<tr>
<td>Hintergrund</td>
<td>prozessen ‘background process’</td>
<td>Hintergrund</td>
</tr>
<tr>
<td>Weltkriegsszenarium ‘world war scenario’</td>
<td>Weltkriegsszenarium</td>
<td>Weltkriegsszenarium</td>
</tr>
</tbody>
</table>

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|arbeits|plätze 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).

<table>
<thead>
<tr>
<th>Gold standard</th>
<th>System</th>
<th>SP_Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>VZDH2014GS/NL</td>
<td>ZvdP2016</td>
<td>94.6</td>
</tr>
<tr>
<td></td>
<td>VZDH2014</td>
<td>91.5</td>
</tr>
<tr>
<td>VZDH2014GS/AF</td>
<td>ZvdP2016</td>
<td>82.9</td>
</tr>
<tr>
<td></td>
<td>VZDH2014</td>
<td>88.3</td>
</tr>
</tbody>
</table>

Table 18.18.: Results for Dutch/Afrikaans compound splitting - external system comparison

Results and Discussion for Dutch and Afrikaans Compound Splitters
18. Multilingual Compound Splitting

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 SP.X). 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.
19. Distributional Semantics

19.1. 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:

\[ \text{You shall know a word by the company it keeps} \]

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 \( V_t \) 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 \( V_c \), 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 a 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 \( D_{\text{sim}} \) maps two target terms \((t_A, t_B)\) on a real value, i.e., \( D_{\text{sim}} : \mathbb{R}^{|C|} \times \mathbb{R}^{|C|} \rightarrow \mathbb{R} \). The commonly used metric for \( D_{\text{sim}} \) between the target terms \( t_A \) and \( t_B \) is the cosine similarity as shown in Formula 19.1.

\[
D_{\text{sim}}(t_A, t_B) = \cosine(\vec{t_A}, \vec{t_B}) = \frac{\sum_{k=1}^{|C|} t_{Ak} \cdot t_{Bk}}{\sqrt{\sum_{i=1}^{|C|} t_{Ai}^2} \cdot \sqrt{\sum_{i=1}^{|C|} t_{Bi}^2}}
\]

### 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 \( D_{\text{sim}} \) 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 \( \textit{Eidotter} \) ‘egg yolk’, the assumption is that \( D_{\text{sim}}(\vec{\textit{Eidotter}}, \vec{\textit{Ei}}) > D_{\text{sim}}(\vec{\textit{Eidotter}}, \vec{\textit{Eid}}) \) and that \( D_{\text{sim}}(\vec{\textit{Eidotter}}, \vec{\textit{Dotter}}) > D_{\text{sim}}(\vec{\textit{Eidotter}}, \vec{\textit{Otter}}) \).

This approach to measure the \( D_{\text{sim}} \) 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 \( \textit{Apfelsaft} \) ‘apple juice’ has a high \( D_{\text{sim}} \) to both constituents (\( \textit{Apfel} \) ‘apple’ and \( \textit{Saft} \) ‘juice’), a non-compositional compound such as \( \textit{Maulwurf} \) ‘mole’ has a very low \( D_{\text{sim}} \) to alleged constituents (i.e., \( \textit{Maul} \) ‘mouth’ and \( \textit{Wurf} \) ‘toss’).

Our procedure of using \( D_{\text{sim}} \) 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 \( D_{\text{sim}} \) 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 \( D_{\text{sim}} \). For example, if \( \textit{Eid} \mid \textit{Otter} \) would be the correct split for \( \textit{Eidotter} \), the interpretation would be non-compositional, because \( \textit{Eid} \) ‘oath’ and \( \textit{Otter} \) ‘otter’ are not related to the intended meaning of \( \textit{Eidotter} \) ‘egg yolk’.

\(^1\)For the sake of simplicity, we use \( D_{\text{sim}} \) 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 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.

<table>
<thead>
<tr>
<th>Ranking score</th>
<th>LSF</th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>14264</td>
<td>Fisch + Zeugnis ‘fish certificate’</td>
<td>✗</td>
</tr>
<tr>
<td>9390</td>
<td>Fisch + Erzeugnis ‘fish product’</td>
<td>✓</td>
</tr>
<tr>
<td>5387</td>
<td>Fischer + Zeugnis ‘fisherman certificate’</td>
<td>✗</td>
</tr>
</tbody>
</table>

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 $D_{sim}$ (cf. Formula 19.1). Table 19.2 shows the $D_{sim}$ values for all candidate splits presented in Table 19.1.

These $D_{sim}$ values are used to predict the degree of semantic similarity between the intended meaning of the target compound and that of its candidate constituents.

---

2Following Weller et al. (2014), we focus on true compounds and ignore non-split options, i.e., atomic analyses.
19.3.3. Distributional Similarity Modes

Besides directly using the \( D_{\text{sim}} \) 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, \( \psi_1, \ldots, \psi_N \).

**Geometric mean** As proposed by Weller et al. (2014), a possible combination of the candidate constituents’ \( D_{\text{sim}} \) values is the geometric mean (GEO), as shown in Formula 19.2.

\[
\text{GEO}(\psi_1 + \cdots + \psi_N) = \sqrt[N]{\prod_{i=1}^{N} D_{\text{sim}}(\Psi, \psi_i)} \tag{19.2}
\]

For example, let \( D_{\text{sim}}(\text{Fischerzeugnis}, \text{Fisch}) \) be 0.455 and \( D_{\text{sim}}(\text{Fischerzeugnis}, \text{Erzeugnis}) \) 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.

<table>
<thead>
<tr>
<th>LSF</th>
<th>GEOscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Fisch} + \text{Erzeugnis} )</td>
<td>( \sqrt{0.46 \cdot 0.10} \approx 0.22 )</td>
</tr>
<tr>
<td>( \text{Fisch} + \text{Zeugnis} )</td>
<td>( \sqrt{0.46 \cdot 0.01} \approx 0.05 )</td>
</tr>
<tr>
<td>( \text{Fischer} + \text{Zeugnis} )</td>
<td>( \sqrt{0.03 \cdot 0.01} \approx 0.01 )</td>
</tr>
</tbody>
</table>

Table 19.3.: Geometric mean \( D_{\text{sim}} \) 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),
19. Semantically Informed Compound Splitting using Shallow Semantics

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 $\Psi$ and the vector sum of modifiers and head ($\sum_{i=1}^{N} \psi_i$), as shown in Formula 19.3.

$$\text{ADD}(\psi_1 + \cdots + \psi_N) = Dsim(\Psi, \sum_{i=1}^{N} \psi_i)$$ (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 $\Psi$ and the vector product of modifiers and head ($\prod_{i=1}^{N} \psi_i$), as shown in Formula 19.4.

$$\text{MULT}(\psi_1 + \cdots + \psi_N) = Dsim(\Psi, \prod_{i=1}^{N} \psi_i)$$ (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.

<table>
<thead>
<tr>
<th>Ranking score</th>
<th>LSF</th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>9390 · 0.22</td>
<td>Fisch + Erzeugnis</td>
<td>✓</td>
</tr>
<tr>
<td>≈ 2034</td>
<td>‘fish product’</td>
<td></td>
</tr>
<tr>
<td>14264 · 0.05</td>
<td>Fisch + Zeugnis</td>
<td>x</td>
</tr>
<tr>
<td>≈ 709</td>
<td>‘fish certificate’</td>
<td></td>
</tr>
<tr>
<td>5387 · 0.01</td>
<td>Fischer + Zeugnis</td>
<td>x</td>
</tr>
<tr>
<td>≈ 70</td>
<td>‘fisherman certificate’</td>
<td></td>
</tr>
</tbody>
</table>

Table 19.4.: Split re-ranking with GEO scores

244
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 \( D_{sim} \) value is 0.

A reason for why it is not possible to compute the \( D_{sim} \) 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 \( D_{sim}(\Psi, \psi_i) \) 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\(^3\).

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.

\(^3\)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 TREE TAGGER. 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 F1-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-occurrents.

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 ($rr_{freq} \cdot ds$). The baseline ($rr_{ds}$) 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 RRds baseline. While the RRfreq·ds re-ranker falls back on the initial frequency-based scores (as discussed in Section 19.3.5), for RRds, there is no information for re-ranking. For comparing RRds with RRfreq·ds and with INIT on a common test set, we additionally DISCarded UnKnown compounds (discUK) for RRfreq·ds and INIT. The third column in Table 19.5 shows the test set sizes after discarding unknown compounds.

<table>
<thead>
<tr>
<th>System</th>
<th>Test set size</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all</td>
<td>discUK</td>
</tr>
<tr>
<td>FF2010</td>
<td>5111</td>
<td>4121</td>
</tr>
<tr>
<td>WH2012</td>
<td>42,172</td>
<td>38,443</td>
</tr>
<tr>
<td>ZvdP2016</td>
<td>52,443</td>
<td>47,951</td>
</tr>
<tr>
<td>TOTAL</td>
<td>52,937</td>
<td>48,008</td>
</tr>
</tbody>
</table>

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 ($\text{rr}_{\text{freq}} \cdot \text{ds}$) improves the initial split ranking ($\text{INIT}$) for at least one SiMode (i.e., GEO).

The baseline of using pure SiMode scores ($\text{rr}_{\text{ds}}$) significantly worsens the initial performance ($\text{INIT}$). 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 $\text{rr}_{\text{ds}}$, is a crucial feature for compound splitting and helps to demote analyses based on typographical errors or unlikely constituent normalization. For example, while $\text{rr}_{\text{freq}} \cdot \text{ds}$ analyzes the compound *Haarwasser* ‘hair tonic’ with the correct and highly frequent modifier *Haar* ‘hair’, $\text{rr}_{\text{ds}}$ 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 $\text{rr}_{\text{ds}}$ 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 $\text{rr}_{\text{freq}} \cdot \text{ds}$, 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@RRFREQ-DS@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 RRFREQ-DS). 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 split 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-branching 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

 constituents *Zinnguss* ‘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.

<table>
<thead>
<tr>
<th>Metric</th>
<th>SP Acc</th>
<th>Norm Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simode</td>
<td>MOD</td>
<td>头</td>
</tr>
<tr>
<td>INIT</td>
<td>88.0</td>
<td>80.3</td>
</tr>
<tr>
<td>RRfreq</td>
<td>87.3</td>
<td>86.7</td>
</tr>
<tr>
<td>UPPER</td>
<td>99.4</td>
<td>95.7</td>
</tr>
</tbody>
</table>

Table 19.6.: Results of split re-ranking for FF2010

When re-ranking (RRfreq-ds) 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 compound
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.

<table>
<thead>
<tr>
<th>Metric</th>
<th>SPAcc</th>
<th>NormAcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiMode</td>
<td>MOD</td>
<td>HEAD</td>
</tr>
<tr>
<td>INIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RR_{FRQ-DS}</td>
<td>98.2</td>
<td>98.2</td>
</tr>
<tr>
<td>UPPER</td>
<td>99.9</td>
<td></td>
</tr>
<tr>
<td>RR_{DS}</td>
<td>96.4</td>
<td>96.5</td>
</tr>
<tr>
<td>RR_{FRQ-DS}</td>
<td>98.4</td>
<td>98.4</td>
</tr>
<tr>
<td>UPPER</td>
<td>99.9</td>
<td></td>
</tr>
</tbody>
</table>

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_{FRQ-DS}) promotes the correct analysis: Zahn | Seide. For the large test set of WH2012, re-ranking (RR_{FRQ-DS}) 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
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_{FREQ}·DS promotes the correct compound split (Denk|ansatz) with the same modifier lemma denken and the head lemma Ansatz ‘approach’. For RR_{FREQ}·DS, all SiModes yield a significant (✓) improvement over the initial split ranking (INIT).

<table>
<thead>
<tr>
<th>Metric</th>
<th>SP Acc</th>
<th>Norm Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiMode</td>
<td>MOD GE HEAD GEO MULT ADD MOD HEAD GEO MULT ADD</td>
<td></td>
</tr>
<tr>
<td>INIT</td>
<td>97.9</td>
<td>88.2</td>
</tr>
<tr>
<td>RR_{FREQ}·DS</td>
<td>98.0✓</td>
<td>98.2✓</td>
</tr>
<tr>
<td>UPPER</td>
<td>99.8</td>
<td>97.4</td>
</tr>
<tr>
<td>INIT</td>
<td>98.1</td>
<td>88.5</td>
</tr>
<tr>
<td>RR_{DS}</td>
<td>94.5✓</td>
<td>94.9✓</td>
</tr>
<tr>
<td>RR_{FREQ}·DS</td>
<td>98.2✓</td>
<td>98.4✓</td>
</tr>
<tr>
<td>UPPER</td>
<td>99.9</td>
<td>97.5</td>
</tr>
</tbody>
</table>

Table 19.8.: Results of split re-ranking for ZvdP2016
19. Semantically Informed Compound Splitting using Shallow Semantics

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

<table>
<thead>
<tr>
<th>Text $T$</th>
<th>Hypothesis $H$</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter studies NLP</td>
<td>Peter is a student</td>
<td>$T \Rightarrow H$ (ENTAILMENT)</td>
</tr>
<tr>
<td>Peter buys a computer</td>
<td>Peter has a wife</td>
<td>$T \not\Rightarrow H$ (NON-ENTAILMENT)</td>
</tr>
<tr>
<td>Mary’s husband sleeps</td>
<td>Mary is single</td>
<td>$T \Rightarrow \neg H$ (CONTRADICTION)</td>
</tr>
</tbody>
</table>

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 Kinderbuch ‘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 $T \not\Rightarrow H$ classification.

(9) a. $T$: *Peter liest ein Buch* ‘Peter reads a book’
   b. $H$: *Peter liest die Bücher*|*regalaufbauanleitung* ‘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*. 

258
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.

    b. $H$: Der {Flug Kapitän} fliegt ein {Flug Zeug} ‘The [aircraft] captain controls an [airplane]’

    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. \( H_1 \checkmark \): Peters {Abend essen} enthielt zwei Ei Dotter
    ‘Peter’s dinner includes two egg yolks’

c. \( H_2 \times \): Peters Abend essen enthielt zwei Eid Otter
    ‘Peter’s dinner includes two oath otters’
While the correct splitting of \textit{Abend|essen} ‘dinner’ and \textit{Ei|dotter} ‘egg yolk’ in $H_1$ leads to a token coverage of $\frac{5}{7} \approx 71\%$ (assuming that \textit{essen} ‘meal’ is aligned to the derived verb \textit{isst} ‘eats’ and \textit{abends} ‘in the evening’ is aligned to the noun \textit{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 \textit{Gurkentisch} ‘cucumber table’ would fail unless it is split into the constituents \textit{Gurke} ‘cucumber’ and \textit{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), Fritzinger 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 preprocessed (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.

On the basis of the multi-level alignment framework, Noh et al. (2015) present a publicly available 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).

---

1[https://github.com/hltfbk/EOP-1.2.1/wiki/AlignmentEDAP1](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 TreeTagger 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\(^2\).

#### 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.

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.

$$P_{T \Rightarrow H} = \frac{T \Rightarrow H \checkmark}{T \Rightarrow H \checkmark + T \not\Rightarrow H \checkmark} \quad (20.2)$$

$$P_{T \not\Rightarrow H} = \frac{T \not\Rightarrow H \checkmark}{T \not\Rightarrow H \checkmark + T \not\Rightarrow H \times} \quad (20.3)$$

$$R_{T \Rightarrow H} = \frac{T \Rightarrow H \checkmark}{T \Rightarrow H \checkmark + T \not\Rightarrow H \checkmark} \quad (20.4)$$

$$R_{T \not\Rightarrow H} = \frac{T \not\Rightarrow H \checkmark}{T \not\Rightarrow H \checkmark + T \not\Rightarrow H \times} \quad (20.5)$$

$$F_{1T \Rightarrow H} = \frac{2 \cdot P_{T \Rightarrow H} \cdot R_{T \Rightarrow H}}{P_{T \Rightarrow H} + R_{T \Rightarrow H}} \quad (20.6)$$

$$F_{1T \not\Rightarrow H} = \frac{2 \cdot P_{T \not\Rightarrow H} \cdot R_{T \not\Rightarrow H}}{P_{T \not\Rightarrow H} + R_{T \not\Rightarrow H}} \quad (20.7)$$

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\(^3\) (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.

\(^3\)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 TREETAGGER, 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 $WH_{2012}$ and $ZvdP_{2016}$ (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 $ZvdP_{2016}$, 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’, \ldots\}. 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 ($ZvdP_{2016}$, FF2010 and $WH_{2012}$), as well as for the addition of a DErivBase lexical aligner.
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 accidently 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. \textit{T}: Chirac brauchte von den \textbf{Wählern} ein neues Mandat für seine \textbf{Regierung} ...  
‘Chirac needed a new mandate for his \textbf{government} from the \textbf{electorate} ...’

b. \textit{H}: \textit{Parlaments|wahlen} führen zur Gründung einer neuen \textbf{Regierung} in \textit{Frankreich}  
‘Parliamentary \textbf{elections} create new \textbf{government} in \textit{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 Fritzinger 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
21. Bottom Line of Compound Splitting

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 \texttt{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 \texttt{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 (∼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 (∼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 (∼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?

⇒ MOPs are designed in a universal string-based matter 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., $^\wedge/\alpha$), 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
Bottom Line of Compound Splitting

in a row, one such consonant is dropped (Stymne and Holmqvist, 2008), e.g., \textit{stopplikt} \textit{→} \textit{stopplikt} ‘stop obligation’. While we are able to model the truncation of a word-final \textit{p} (i.e., \textit{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 \textasciitilde and $, such operations can be applied unambiguously. However, for the word-internal operations, the concrete replacement position is underspecified. This underspecification is a deliberate feature for generalizing over various constituents. For example, the Umlautung MOP \textit{u$/$ü} can be applied both to \textit{Buch} ‘book’ (leading to \textit{Bücher}, i.e., Umlautung at the second position) and to \textit{Bruder} ‘brother’ (leading to \textit{Brüder}, i.e., Umlautung at the third position). However, the underspecified replacement positions also leads to ambiguity when having several word-internal substrings that match with the source side of a replacement. For example, for the constituent \textit{Suppenhuhn} ‘boiling hen’ (as in the compound \textit{Suppenhühnerverkauf} ‘boiling hen sale’), the Umlautung MOP \textit{u$/$ü} can be applied at the first or second \textit{u}. By convention, this kind of ambiguity is resolved by selecting the last replacement position as default, as described in Section 17.3.

\textbf{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 \textbf{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 (\textbf{RQ\_2-B-i}) and language-specific approaches (\textbf{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 Fritzinger 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, $k_{\text{gold}}$). 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.
21. Bottom Line of Compound Splitting

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 (SPAcc) 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?
21. Bottom Line of Compound Splitting

⇒ 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\textsubscript{FREQ-DS} shows a consistent improvement for all inspected compound splitters, the RR\textsubscript{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
21. Bottom Line of Compound Splitting

tonic’ with the correct and highly frequent modifier \textit{Haar} ‘hair’. In contrast, \textit{RR}_{\text{ds}} \text{ would select the morphologically plausible but yet unlikely and infrequent verbal modifier \textit{haaren} ‘to molt’, which happens to have the higher Dsim to \textit{Haarwasser}. On the other hand, binary splitting of 3NCs (i.e., bracketing) benefits from Dsim information. For example, using re-ranking by \textit{RR}_{\text{FREQ-DS}}, the wrong \textit{compound split} \textit{Arbeitsplatzmangel} ‘labor [lack of space]’ (top-ranked in \textit{INIT}) is corrected to \textit{Arbeitsplatzmangel} ‘job scarcity’ in \textit{RR}_{\text{FREQ-DS}}.

\textbf{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 (\textit{SiModes}): modifier (\textit{MOD}) and head (\textit{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 \textit{FF2010}, \textit{MOD} is working best. In contrast, for \textit{WH2012}, \textit{MOD} and \textit{HEAD} perform equally for \textit{SPAcc}, whereas \textit{MOD} outperforms \textit{HEAD} in \textit{NormAcc}. And for \textit{ZvdP2016}, \textit{HEAD} outperforms \textit{MOD} for \textit{SPAcc} and vice versa for \textit{NormAcc}.

\textbf{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 \textit{(LEFT class baseline)}, as will be discussed in Part E) and an ambiguous head. For example, the 3NC \textit{Block\text{"{f}"{o}t\text{"{e}n}\text{"{s}"{p}"{i}\text{"{e}l}er} ‘recorder player’, where the complex head, \textit{Fl"{o}tenspieler} ‘flute player’, is more distributionally similar to the compound than the simplex head \textit{Spieler} ‘player’. In contrast, the complex modifier \textit{Block\text{"{f}"{o}t\text{"{e}n} ‘recorder} is more similar to the compound than the simplex modifier \textit{Block} ‘block’. Therefore, the \textit{SiMode} \textit{HEAD} is not as beneficial as \textit{MOD}.

\textbf{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 \textit{Energie\text{"{s}"{p}"{a}\text{"{r}m} \text{"{a}mpe} ‘energy-saving bulb’, the simplex modifier, \textit{Energie}, is more similar to the compound than the complex (phrasal) modifier, \textit{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 $\text{Norm}_F$ but worst in $\text{SP}_F$. For the M2006GS gold standard, ZvdP2016 is worst in $\text{Norm}_F$, whereas it outperforms FF2010 in $\text{SP}_F$.

What methods perform best with respect to split point determination? For the discipline of split point determination (measured in $\text{SP}_F$), 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 $\text{Norm}_F$), 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:** How 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 **N-ary 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

\[
\text{Langsamfahrstelle} \\
\text{temporary speed restriction} \\
\text{langsam} \quad \text{fahren} \quad \text{Stelle} \\
\text{slowly} \quad \text{drive} \quad \text{section}
\]

Figure 21.1.: Example of a ternary split tree structure for \text{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., \text{stop|plikt} \rightarrow \text{stopp 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 lookahead operator. For example, \( <p_1, p$/\, p_2> \), 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/\ddot{a}:$/\text{er}$ (which is valid for nouns like \textit{Mann} ‘man’) can be falsely triggered for the compound \textit{Läuferteam} ‘runner’s team’, leading to the nominal modifier lemma \textit{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., \textit{Kindheit} ‘childhood’: while \textit{Kindheit} gets s-suffixed as a modifier (as in \textit{Kindheits|erinnerung} ‘childhood memory’), \textit{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(Eidotter, Dotter) vs. cos(Eidotter, 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(Ei, Dotter) vs. cos(Ei, 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 SPF 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 entailment relation in Example 10 could be revealed. In Example 17, the modifier Auto in $H$ is a hyponym of Mercedes in $T$ and the head Fahrer in $H$ is a derivation of fährt (or fahren) in $T$. 

288
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\(^1\) (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?

\(^1\)http://nlp.uned.es/clef-qa/ave/
21. Bottom Line of Compound Splitting
Part E.

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.

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.
22. Introduction to Compound Parsing

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 energyA efficiencyB actionC plansD can be paraphrased in German as AktionspläneD für EnergieeffizienzA (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.
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.

\textbf{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.

\textbf{RQ\_2-A-v:} How useful is the proposed AWD metric?

\textbf{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 bigA priceB labelC can be parsed using the German NP großesA PreisschildB.

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-branched 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:

\[
\text{AdjMod}(A \ B \ C) = \begin{cases} 
\text{LEFT} & \text{if } AM(A, B) > AM(B, C) \\
\text{UNKNOWN} & \text{if } AM(A, B) = AM(B, C) \\
\text{RIGHT} & \text{if } AM(A, B) < AM(B, C)
\end{cases}
\] (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-branched structure: both A and B modify C.
23. Related Work on Compound Parsing

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:

$$\text{DepMod}(A \ B \ C) = \begin{cases} \text{LEFT} & \text{if } AM(A, B) > AM(A, C) \\ \text{UNKNOWN} & \text{if } AM(A, B) = AM(A, C) \\ \text{RIGHT} & \text{if } AM(A, B) < AM(A, C) \end{cases}$$ (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
23. Related Work on Compound Parsing

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\(_W\_A\)** 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 been 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).

\[
Pr(\text{RIGHT}) = Pr((A|B) \rightarrow C|C)
\]
\[
Pr(\text{LEFT}) = Pr(A \rightarrow B|B)
\]
\[
\text{parse}(A \ B \ C) = \begin{cases} 
\text{LEFT} & \text{if } Pr(\text{LEFT}) > Pr(\text{RIGHT}) \\
\text{UNKNOWN} & \text{if } Pr(\text{LEFT}) = Pr(\text{RIGHT}) \\
\text{RIGHT} & \text{if } Pr(\text{LEFT}) < Pr(\text{RIGHT}) 
\end{cases}
\tag{23.4}
\]

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 \( w_j \)

\(\beta\): \( freq(w_i \bar{w}_j) \) frequency of bigrams starting with \( w_i \) but not ending on \( w_j \)

\(\gamma\): \( freq(\bar{w}_i \ w_j) \) frequency of bigrams not starting with \( w_i \) but ending on \( w_j \)

\(\delta\): \( freq(\bar{w}_i \bar{w}_j) \) 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).

\[
\chi^2(w_i, w_j) = \frac{N \cdot (\alpha \cdot \delta - \beta \cdot \gamma)^2}{(\alpha + \gamma) \cdot (\beta + \delta) \cdot (\alpha + \beta) \cdot (\gamma + \delta)}
\tag{23.5}
\]

The parsing decision works straightforward as given in Formula (23.6).

\[
\text{parse}(A \ B \ C) = \begin{cases} 
\text{LEFT} & \text{if } \chi^2(A, B) > \chi^2((A|B), C) \\
\text{UNKNOWN} & \text{if } \chi^2(A, B) = \chi^2((A|B), C) \\
\text{RIGHT} & \text{if } \chi^2(A, B) < \chi^2((A|B), C) 
\end{cases}
\tag{23.6}
\]
23. Related Work on Compound Parsing

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.

\[
\begin{align*}
\text{AMBIG}(w) &= \text{number of thesaurus categories of } w \\
\text{COUNT}(w_1, w_2) &= \text{number of occurrences of } w_1, w_2 \text{ in the training data} \\
\text{FREQ}(t_1, t_2) &= \sum_{w_1 \in t_1} \sum_{w_2 \in t_2} \frac{\text{COUNT}(w_1, w_2)}{\text{AMBIG}(w_1) \cdot \text{AMBIG}(w_2)} \\
\text{CA}(t_1, t_2) &= \frac{\text{FREQ}(t_1, t_2)}{\sum_{i} \text{FREQ}(t_1, i) \cdot \sum_{i} \text{FREQ}(i, t_2)} (23.7)
\end{align*}
\]

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.

<table>
<thead>
<tr>
<th>LEFT</th>
<th>RIGHT</th>
<th>SEMIND</th>
<th>ERROR</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>81</td>
<td>35</td>
<td>29</td>
<td>308</td>
</tr>
<tr>
<td>52.9%</td>
<td>26.3%</td>
<td>11.4%</td>
<td>9.4%</td>
<td>100%</td>
</tr>
<tr>
<td>163</td>
<td>81</td>
<td>35</td>
<td>#</td>
<td>279</td>
</tr>
<tr>
<td>58.4%</td>
<td>29.0%</td>
<td>12.5%</td>
<td>#</td>
<td>100%</td>
</tr>
<tr>
<td>163</td>
<td>81</td>
<td>#</td>
<td>#</td>
<td>244</td>
</tr>
<tr>
<td>66.8%</td>
<td>33.2%</td>
<td>#</td>
<td>#</td>
<td>100%</td>
</tr>
</tbody>
</table>

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
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: \( \text{freq}(w_1n_1n_2w_4) \) 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: \( \text{freq}(n_1w_1\ldots w_i n_2) \). Following Lauer and Dras (1994), Lauer (1995a) defines a parameter representing the degree of acceptability as given in Formula 23.8.

\[
Pr(t_1 \rightarrow t_2) = \frac{1}{\mu} \sum_{w_1 \in t_1 \atop w_2 \in t_2} \frac{\text{COUNT}(w_1, w_2)}{\text{AMBIG}(w_1) \cdot \text{AMBIG}(w_2)} \tag{23.8}
\]

where \( \mu = \sum_{w_1 \in N \atop w_2 \in t_2} \frac{\text{COUNT}(w_1, w_2)}{\text{AMBIG}(w_1) \cdot \text{AMBIG}(w_2)} \).

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 ratio equal to 1, the LEFT class baseline is used.

\[
R_{\text{AdjMod}} = \frac{\sum_{t_1 \in \text{cats}(w_i)} Pr(t_1 \rightarrow t_2)}{\sum_{t_1 \in \text{cats}(w_i)} Pr(t_1 \rightarrow t_3)} \tag{23.9}
\]

\[
R_{\text{DepMod}} = \frac{\sum_{t_1 \in \text{cats}(w_i)} Pr(t_1 \rightarrow t_2) \cdot Pr(t_2 \rightarrow t_3)}{\sum_{t_1 \in \text{cats}(w_i)} Pr(t_1 \rightarrow t_3) \cdot Pr(t_2 \rightarrow t_3)}
\]

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.
23. Related Work on Compound Parsing

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\(^1\) 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)

\(^{1}\)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 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>2</sub> from the brain<sub>4</sub> stem<sub>3</sub> pointing to LEFT), (2) copula paraphrases (as in office<sub>5</sub> building<sub>6</sub> that/which is a skyscraper<sub>7</sub> indicating LEFT) and (3) verbal paraphrases (as in pain<sub>6</sub> associated with arthritis<sub>4</sub> migraine<sub>4</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* → *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* → *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* → *organization*)

5. **Nominalization:** this feature indicates whether the constituent is a nominalization.

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 SemRe ls 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 SemRel(physics winter school) = SemRel(physics school) $\neq$ SemRel(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 Ngram 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):

   \[ \cdots \ V \ W \ X \ Y \ Z \]

2a. If there is evidence for a RIGHT-branched XYZ, Y and Z are merged and the window introduces the constituent left to X:

   \[ \cdots \ V \ W \ X \ YZ \]

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:
23. Related Work on Compound Parsing

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).

3. 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:

\[ \underline{UVWWXYZ} \]

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-branchered.

- 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*:

313
23. Related Work on Compound Parsing

Knowing the 2NC soup bowl, there is a LEFT-branching evidence and thus the window moves left:

\[
\text{wooden French onion soup bowl handle}
\]

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:

\[
\text{wooden French onion soup bowl handle}
\]

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:

\[
\text{wooden French [onion soup] bowl handle}
\]

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:

\[
\text{wooden [French [onion soup]] bowl handle}
\]

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:

\[
\text{wooden [[French [onion soup]] bowl] handle}
\]

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
system already knows the NP \textit{wooden handle} and thus the final bracketing is \textit{right}, leading to the \textit{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 \textit{[soap opera] stars} and \textit{[television producers]} and \textit{[movie and television] producers], where the last three words \textit{and television producers} are structured differently.

\textbf{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 \textit{Air Force contract} is originally represented as:

\begin{verbatim}
(NP
  (NNP Air) (NNP Force) (NN contract))
\end{verbatim}
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$_1$-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$_1$-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$_1$-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)\(^2\) 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 Ngram 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

\(^2\)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 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_A development_B aid_C projects_D being aligned to the Italian progetti_D ecclesiastici_A di aiuti_C allo sviluppo_B, 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

---

$^3$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 $\text{prep}: w_1 \text{ prep } w_2$, where $\text{prep} \in \{about, at, by, for, from, in, of, on, to, with\}$ 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 $\text{conj}: w_1 \text{ conj } w_2$, where $\text{conj} \in \{or, and, nor\}$. 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 ‘$\text{DET } w_1 w_2$’, where $\text{DET} \in \{a, an, the\}$ (i.e., a determiner introducing the underlying word pair), ‘$w_1 \text{ plural}(w_2)$’, where $\text{plural}(\ldots)$ denotes the plural form (i.e., the underlying word pair with a pluralized second word) and ‘$\text{DET } w_1 \text{ plural}(w_2)$’ 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 gains 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 MausSubjObj jagt die KatzeSubjObj 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$_{w_1}$ and mortar$_{w_2}$ attacks$_h$ includes an ellipsis for rocket$_{w_1}$ attacks$_h$, the coordination asbestos$_{w_1}$ and polyvinyl$_{w_2}$ chloride$_h$ does not imply asbestos$_{w_1}$ chloride$_h$. 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
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 von Menschenrechten, 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.

<table>
<thead>
<tr>
<th>USP tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>complex noun</td>
</tr>
<tr>
<td>ADJ</td>
<td>adjective</td>
</tr>
<tr>
<td>SN</td>
<td>simplex noun</td>
</tr>
<tr>
<td>FC</td>
<td>sequence of function words</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>German</th>
<th>Swedish</th>
<th>French</th>
<th>Italian</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>CN</td>
<td>SN FC SN FC SN</td>
<td>SN FC SN FC SN</td>
</tr>
<tr>
<td>ADJ CN</td>
<td>ADJ CN</td>
<td>SN FC SN ADJ</td>
<td>SN FC SN ADJ</td>
</tr>
<tr>
<td>SN FC CN</td>
<td>SN</td>
<td>SN FC SN</td>
<td>SN ADJ</td>
</tr>
<tr>
<td>SN</td>
<td>ADJ SN</td>
<td>SN ADJ</td>
<td>SN FC SN</td>
</tr>
<tr>
<td>ADJ SN</td>
<td>SN SN</td>
<td>SN SN FC SN</td>
<td>SN SN FC SN</td>
</tr>
<tr>
<td>SN SN</td>
<td>SN FC ADJ SN</td>
<td>SN SN SN</td>
<td>SN ADJ FC SN</td>
</tr>
<tr>
<td>SN FC ADJ SN</td>
<td>PC CN</td>
<td>SN FC SN SN</td>
<td>SN SN SN</td>
</tr>
<tr>
<td>SN SN SN</td>
<td>PC SN</td>
<td>SN ADJ FC SN</td>
<td>SN FC SN SN</td>
</tr>
<tr>
<td>CN FC SN</td>
<td>SN SN</td>
<td>SN SN</td>
<td>SN SN ADJ</td>
</tr>
<tr>
<td>CN FC CN</td>
<td>SN VB SN</td>
<td>SN SN ADJ</td>
<td>SN SN ADJ</td>
</tr>
</tbody>
</table>

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).
24. Pilot Study using Aligned Phrase Patterns

<table>
<thead>
<tr>
<th>Aligned Phrase Pattern</th>
<th>Assigned structure class</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ CN</td>
<td>RIGHT</td>
</tr>
<tr>
<td>CN FC SN</td>
<td>RIGHT</td>
</tr>
<tr>
<td>SN FC CN</td>
<td>LEFT</td>
</tr>
<tr>
<td>SN FC ADJ SN</td>
<td>LEFT</td>
</tr>
<tr>
<td>SN ADJ FC SN</td>
<td>RIGHT</td>
</tr>
<tr>
<td>ADJ SN FC SN</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>

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)
24. Pilot Study using Aligned Phrase Patterns

<table>
<thead>
<tr>
<th>APP</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ CN</td>
<td>German (STATELICHE STEUERAUFSCHTSBEHÖRDENCN): staatliche\textsubscript{ADJ} Steueraufsichtsbehörden\textsubscript{CN} ‘state tax inspectorates’</td>
</tr>
<tr>
<td>CN FC SN</td>
<td>German (SATZMARKT\textsubscript{CN} FÜR\textsubscript{FC} FAHRZEUGE\textsubscript{SN}) (lit: ‘sales market for cars’) ‘car sales market’</td>
</tr>
<tr>
<td>SN FC CN</td>
<td>Dutch (\textit{METHODE\textsubscript{SN} VOOR\textsubscript{FC} GEBORTEBEPERKING\textsubscript{CN}}) (lit: ‘method for birth control’) ‘birth control method’</td>
</tr>
<tr>
<td>SN FC ADJ SN</td>
<td>Swedish (\textit{BROTTETE\textsubscript{SN} MOT\textsubscript{FC} MÄNSKLI\textsubscript{ADJ} RÄTTIGHETERNA\textsubscript{SN}}) (lit: ‘abuses of human rights’) ‘human rights abuses’</td>
</tr>
<tr>
<td>SN ADJ FC SN</td>
<td>Spanish (\textit{CONSUMO\textsubscript{SN} FINAL\textsubscript{ADJ} DE\textsubscript{FC} ENERGIA\textsubscript{SN}}) (lit: ‘consumption final of energy’) ‘energy end consumption’</td>
</tr>
<tr>
<td>ADJ SN FC SN</td>
<td>Danish (\textit{GEMENSNITLIGE\textsubscript{ADJ} OVERFÆRSEL\textsubscript{SN} AF\textsubscript{FC} DATA\textsubscript{SN}}) (lit: ‘average transfer of data’) ‘data transfer rate’</td>
</tr>
</tbody>
</table>

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 \textit{ADJ SN FC SN})

The assignment of the structure class relies on the assumption of \textbf{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 \textit{right}-branched structure) and so for a complex modifier or simplex head (meaning a \textit{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 \textbf{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
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. $\text{[ ]} \leftarrow \text{Structures}$
2. for all aligned support languages $l_i \in L$ do
   3. $\text{Structure}_i \leftarrow \tau(APP_i)$
   4. if $\text{Structure}_i \neq \text{UNKNOWN}$ then
   5. $\text{Structures} \leftarrow \text{Structures} + [\text{Structure}_i]$
   6. end if
   7. end for
8. return $\max(\text{Structures})$

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.

(18) Dutch: $\text{stabile}_1 \text{ wisselkoersen}_2$

$\text{stable}_1 \text{ } \{\text{exchange rate}_2\}$

"$\{\text{exchange rate}_2\}_2 \text{ stability}_1$"

(19) German: $\text{Resolutions}_1 | \text{entwurf}_2$

$\{\text{resolution}_1 \text{ draft}_2\}$

"draft_2 resolution_1"

For taking into account constituent swapping during APPP, a word alignment-based interpretation of the complex unit is added to the compound parser. Therefor, 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. $\left[ \right] \leftarrow$ Structures
2. for all support languages $l_i \in L$ do
3. \hspace{1em} default$_i \leftarrow \tau(\text{APP}_i)$
4. \hspace{1em} if default$_i \neq \text{UNKNOWN}$ then
5. \hspace{2em} if $\text{CU}(\Psi, \text{APP}_i) \sim \{B, C\}$ or $\text{CU}(\Psi, \text{APP}_i) \sim \{A, C\}$ then
6. \hspace{3em} Structures $\leftarrow$ Structures + [RIGHT]
7. \hspace{2em} else if $\text{CU}(\Psi, \text{APP}_i) \sim \{A, B\}$ then
8. \hspace{3em} Structures $\leftarrow$ Structures + [LEFT]
9. \hspace{2em} else if $\text{CU}(\Psi, \text{APP}_i) \sim \{A, B, C\}$ then
10. \hspace{3em} Structures $\leftarrow$ Structures \{indicates word alignment error\}
11. \hspace{1em} else
12. \hspace{2em} Structures $\leftarrow$ Structures + [default$_i$] \{otherwise, use the default\}
13. \hspace{1em} 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, 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$_i$: if the target constituent sets $\{B, C\}$ or $\{A, C\}$ 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.

---

1. In 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. Pilot Study using Aligned Phrase Patterns

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEFT baseline</td>
<td>71.1%</td>
</tr>
<tr>
<td>APPP</td>
<td>89.0%</td>
</tr>
<tr>
<td>APPPWA</td>
<td>91.6%</td>
</tr>
</tbody>
</table>

Table 24.5: Parsing results for APPP and APPPWA

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\(^2\) better than the LEFT class baseline.

The compound parsing using APPs and word alignment information (APPPWA) leads to the best results with 91.6% accuracy, outperforming both the LEFT class baseline and regular APPP.

\(^2\)Approximate randomization test (Yeh, 2000), \(p < 5\%\)
24. Pilot Study using Aligned Phrase Patterns

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 für Energieeffizienz ‘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 A violations B aligned to the Italian sentence fragment che le violazioni C gravi e sistematiche dei diritti umani C. 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\_WA, 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\(^1\) of $c_i$ (i.e., the cross-lingual equivalents) in $l$. If $c_i$ is a complex constituent composed of the atomic constituents $c_{i_1}, \ldots, c_{i_n}$, $\text{AWS}_l(c_i)$ 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

\(^1\)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.
pos\( (w_i) \) is the sentence position (e.g., a token counter starting by 1) of an aligned word \( w_i \).

\[
\text{AWS}_i(c_i) = \begin{cases} 
\{w_{i,x} | \text{content word}(w_{i,x}) \land pos(w_i) = x \land c_i \Rightarrow l w_i \} & \text{if } c_i \text{ is atomic} \\
\bigcup_{c_k \in c_i} \text{AWS}_i(c_k) & \text{if } c_i \text{ is complex} \end{cases} \tag{25.1}
\]

For two target constituents \( c_i, c_j \in \Psi \), the AWD\(_l\) between them is defined as given in Formula 25.2.

\[
\text{AWD}_l(c_i, c_j) = \min_{x \in \text{AWS}_i(c_i), y \in \text{AWS}_j(c_j)} |pos(x) - pos(y)| \tag{25.2}
\]

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 \( \ldots \text{che}_1 \text{ le}_2 \text{ violazioni}_3 \text{ gravi}_4 \text{ sistematiche}_6 \text{ dei}_7 \text{ diritti}_8 \text{ umani}_9 \), the \( \text{AWS}_i(\text{human}) \) is \( \{ \text{uman}_9 \} \), the \( \text{AWS}_i(\text{rights}) \) is \( \{ \text{diritti}_8 \} \) and the \( \text{AWS}_i(\text{violations}) \) is \( \{ \text{violazioni}_3 \} \), 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:

\[
\text{AWD}_i(\text{human, rights}) = \min_{x \in \text{AWS}_i(\text{human}), y \in \text{AWS}_i(\text{rights})} |pos(x) - pos(y)| \\
= \min_{x \in \{ \text{uman}_9 \}, y \in \{ \text{diritti}_8 \}} |pos(x) - pos(y)| \\
= |9 - 8| = 1
\]

\[
\text{AWD}_i(\text{human, violations}) = \min_{x \in \text{AWS}_i(\text{human}), y \in \text{AWS}_i(\text{violations})} |pos(x) - pos(y)| \\
= \min_{x \in \{ \text{uman}_9 \}, y \in \{ \text{violazioni}_3 \}} |pos(x) - pos(y)| \\
= |9 - 3| = 6
\]
\[ \text{AWD}_{it}(\text{rights, violations}) = \min_{x \in \text{AWS}_{it}(\text{rights}), \ y \in \text{AWS}_{it}(\text{violations})} |\text{pos}(x) - \text{pos}(y)| \]
\[ = \min_{x \in \{\text{diritti}_8\}, \ y \in \{\text{violazioni}_3\}} |\text{pos}(x) - \text{pos}(y)| \]
\[ = |8 - 3| = 5 \]

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 $\Psi$: $c_1, \ldots, c_k$

**Input:** AWS$_l$(ci) for the support language $l$ and for all constituents $c_i \in$ BOTTOM

1: $\text{CONs} \leftarrow c_1, \ldots, c_k$ \{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$_l$(ci) = AWS$_l$(cm) $\cup$ AWS$_l$(cm+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: CONs $\leftarrow$ 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\}

In each iteration, the pair of adjacent constituents having the smallest AWD is determined. For this pair, the corresponding AWSs are unified and the constituents are replaced by the merged constituent. Finally, the merged constituent is stored in CONs. The output of the algorithm is a parse tree which is constructed from CONs.

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(ci), i.e., there is no alignment from ci 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, européenne chargée de la sécurité du transport aérien 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](image)

*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 gaspijpleiding met dubbele pijp 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 {gaspipjpleiding$_{22}$} and AWS$_{nl}$(pipeline) is also {gaspipjpleiding$_{22}$}.
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.

![Parse tree for twin pipe undersea gas pipeline]

**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 ENCD 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 ENCD, 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 \( P_{\text{noun}}(\text{word}) = P(\text{noun} \mid \text{word}) \). It is trained on the English WIKIPEDIA tagged by TREE TAGGER (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
\( P_{\text{noun}}(N_i) < \theta \) or \( P_{\text{noun}}(w_j) \geq \theta \), 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\(^2\) 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
\( P_{\text{noun}}(\text{word}) \)

**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).

\(^2\)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 \textsc{left}/\textsc{right} consensus, comprising 278 compound tokens.

\textbf{Evaluation Measures}

We measure the \textsc{left}/\textsc{right} classification accuracy ($\text{Acc}_\Omega$) for a set of 3NC tokens $\Omega$ as shown in Formula 25.3, i.e., the number of correctly \textsc{left}- or \textsc{right}-branched 3NCs divided by size of $\Omega$.

\[
\text{Acc}_\Omega = \frac{freq(\textsc{left}\,\checkmark) + freq(\textsc{right}\,\checkmark)}{|\Omega|}
\]  

(25.3)

The coverage ($\text{Cov}$) of a method is shown in Formula 25.4, i.e., the number of all assigned \textsc{left}/\textsc{right} labels divided by all 3NC tokens in the full dataset described above.

\[
\text{Cov} = \frac{freq(\textsc{left}) + freq(\textsc{right})}{14,941}
\]  

(25.4)

We consider an optimal compound parsing method to have the best trade-off between $\text{Acc}_\Omega$ and $\text{Cov}$. Therefore, we used the harmonic mean (harmonic) between these measures.

\textbf{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 bigram frequencies derived from the English part of \textsc{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 → \( \chi^2 \) and DBUP → \( \chi^2 \), respectively.

Finally, we compare with the LEFT class baseline.

Results

Table 25.1 presents the coverage of each system.

<table>
<thead>
<tr>
<th>System</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBUP_token / DBUP_type</td>
<td>87.9% / 91.2%</td>
</tr>
<tr>
<td>DBUP_type → ( \chi^2 )</td>
<td>100%</td>
</tr>
<tr>
<td>( \chi^2 )</td>
<td>100%</td>
</tr>
<tr>
<td>APPP_token / APPP_type</td>
<td>29.9% / 48.1%</td>
</tr>
<tr>
<td>APPP_type → ( \chi^2 )</td>
<td>100%</td>
</tr>
<tr>
<td>LEFT class baseline</td>
<td>100%</td>
</tr>
</tbody>
</table>

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\_type). As expected, our pattern-based approach does not cover more than 48.1% (APPP\_type). The statistical \( \chi^2 \) method and the back-off models can process all 3NCs in the dataset. The fact that DBUP\_type misses 8.8% of the dataset is mainly due to equal AWDs between the target constituents. For example, crisis\(_A\) resolution\(_B\) mechanism\(_C\) is only aligned to closed compounds, such as the Swedish krislösningsmekanism (i.e., \( \text{AWD}_{sv}(A,B) = \text{AWD}_{sv}(B,C) = 0 \)), or to nouns separated by one preposition, such as the Spanish mecanismo de resolución de crisis ‘mechanism of resolution of crisis’ (i.e., \( \text{AWD}_{es}(A,B) = \text{AWD}_{es}(B,C) = 2 \)).

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\(^3\)

\(^3\)Approximate randomization test (Yeh, 2000), \( p < 5\% \)
25. Compound Parsing Methods using Aligned Word Distance

<table>
<thead>
<tr>
<th>System</th>
<th>Acc&lt;sub&gt;com&lt;/sub&gt;</th>
<th>harmonic&lt;sub&gt;(com)&lt;/sub&gt;</th>
<th>com</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBUP&lt;sub&gt;token&lt;/sub&gt; / DBUP&lt;sub&gt;type&lt;/sub&gt;</td>
<td>94.4% / 94.4%</td>
<td>91.0% / 92.8%</td>
<td>270</td>
</tr>
<tr>
<td>APPP&lt;sub&gt;token&lt;/sub&gt; / APPP&lt;sub&gt;type&lt;/sub&gt;</td>
<td>87.8% / 87.2%</td>
<td>44.6% / 62.0%</td>
<td>180</td>
</tr>
<tr>
<td>DBUP&lt;sub&gt;type&lt;/sub&gt;</td>
<td>94.6%</td>
<td>92.9%</td>
<td>184</td>
</tr>
<tr>
<td>APPP&lt;sub&gt;type&lt;/sub&gt;</td>
<td>86.4%</td>
<td>61.8%</td>
<td></td>
</tr>
<tr>
<td>DBUP&lt;sub&gt;type&lt;/sub&gt; → χ&lt;sup&gt;2&lt;/sup&gt;</td>
<td>94.1%</td>
<td>92.6%</td>
<td>273</td>
</tr>
<tr>
<td>APPP&lt;sub&gt;type&lt;/sub&gt; → χ&lt;sup&gt;2&lt;/sup&gt;</td>
<td>87.9%</td>
<td>93.6%</td>
<td></td>
</tr>
<tr>
<td>χ&lt;sup&gt;2&lt;/sup&gt;</td>
<td>87.9%</td>
<td>93.6%</td>
<td></td>
</tr>
<tr>
<td>LEFT class baseline</td>
<td>80.9%</td>
<td>89.4%</td>
<td></td>
</tr>
</tbody>
</table>

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 χ<sup>2</sup> in accuracy but is inferior in harmonic<sub>(com)</sub>; here, the higher coverage of χ<sup>2</sup> 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<sub>(com)</sub> 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<sub>(com)</sub> 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).

<table>
<thead>
<tr>
<th>Language family</th>
<th>Acc&lt;sub&gt;com&lt;/sub&gt;</th>
<th>Cov</th>
<th>harmonic&lt;sub&gt;(com)&lt;/sub&gt;</th>
<th>com</th>
</tr>
</thead>
<tbody>
<tr>
<td>Romance</td>
<td>86.6%</td>
<td>86.2%</td>
<td>86.4%</td>
<td>201</td>
</tr>
<tr>
<td>Germanic</td>
<td>94.0%</td>
<td>68.0%</td>
<td>78.9%</td>
<td></td>
</tr>
</tbody>
</table>

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 DBUP\textsubscript{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 SN FC SN ADJ as in the English 3NC state health service being aligned to the Portuguese serviços de saúde estatais (lit.: [service\textsubscript{pl} of health\textsubscript{sg}] state\textsubscript{adj}). 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\(^4\) 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\textsubscript{pl} de pêche mondial\textsubscript{adj} (lit: ‘quotas\textsubscript{pl} of fishing\textsubscript{sg} world\textsubscript{adj,pl}’).

### 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

\(^4\text{z-test for proportions; } p < 5\%\)
Compound. For example, the English 4NC book price fixing schemes is aligned to the German closed compound Buchpreisbindungsgesetze, 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.

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).
25. Compound Parsing Methods using Aligned Word Distance

In Section 25.3.3, the NTAP method is more fine-grained and considers all subtrees\(^5\) 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 \(l\), if for each node \(N \in PT\), the AWD\(_l\) between the target constituents related to the joined daughter nodes of \(N\) is smaller than (or equal to) the AWD\(_l\) between the constituents related to \(N\) and the sister node of \(N\).

<table>
<thead>
<tr>
<th>Air traffic control</th>
<th>Air traffic control</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWD(_nl) = 3</td>
<td>AWD(_nl) = 0 (\frac{1}{2})</td>
</tr>
<tr>
<td>air traffic</td>
<td>air traffic</td>
</tr>
<tr>
<td>AWD(_nl) = 0</td>
<td>AWD(_nl) = 3</td>
</tr>
<tr>
<td>control</td>
<td>control</td>
</tr>
<tr>
<td>air</td>
<td>traffic</td>
</tr>
</tbody>
</table>

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

---

\( A \) 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: \( \text{Trees} \leftarrow \text{generate all possible binary parse trees for } \Psi \)

2: \( \text{for } t \in \text{Trees} \text{ do} \)

3: \( \text{annotate all nodes } N_i \text{ in } t \text{ with AWDs for the support language } l_i \)

4: \( \text{if } \exists N[AWD_{l_i} \text{ mother}(N).AWD_{l_i}] \text{ then} \)

5: \( t \leftarrow \text{INVALID} \)

6: \( \text{end if} \)

7: \( \text{end for} \)

8: \( \text{return } \{t \in \text{Trees } t \text{ is not INVALID}\} \)

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 \( \text{Cat}_{k-1} \) possible binary trees, where the formula for \( \text{Cat}_n \) is given in Formula 3.1, repeated as Formula 25.5.

\[
\text{Cat}_n = \frac{(2n)!}{(n+1)! \cdot n!} \quad (25.5)
\]

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 \( \text{AWD}_{l_i} \) (line 3) according to the formula shown in Formula 25.6, i.e., leaf nodes are annotated with \( \text{AWD}_{l_i} = 0 \) and non-terminal nodes are annotated with the \( \text{AWD}_{l_i} \) between their left and right daughters’ constituent.
Table 25.4.: Number of possible binary parse trees for compounds with $k$ constituents

<table>
<thead>
<tr>
<th>Compound size $k$</th>
<th>Binary trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>$k$</td>
<td>$Cat_{k-1}$</td>
</tr>
</tbody>
</table>

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
25. Compound Parsing Methods using Aligned Word Distance

(a) energy efficiency action plan
   \( \text{AWD}_{de} = 2 \)
   \( \text{AWD}_{es} = 1 \)

(b) energy efficiency action plan
   \( \text{AWD}_{de} = 2 \)
   \( \text{AWD}_{es} = 1 \)

(c) energy efficiency action plan
   \( \text{AWD}_{de} = 2 \)
   \( \text{AWD}_{es} = 2 \)

350
{for the} efficiency of energy’ and the Spanish equivalent plan de acción de eficiencia energética ‘plan of action of efficiency energy' adj’, 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.

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\textsubscript{A} income\textsubscript{B} stabilisation\textsubscript{C} instrument\textsubscript{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\textsubscript{B} til stabilisering\textsubscript{C} af bedrifternes\textsubscript{A} indkomster\textsubscript{B}, to the German equivalent Instrument\textsubscript{B} zur Stabilisierung\textsubscript{C} der landwirtschaftlichen\textsubscript{A} Einkommen\textsubscript{B}, to the Swedish equivalent instrument\textsubscript{B} för stabilisering\textsubscript{C} av jordbruksinkomsterna\textsubscript{A}, to the Spanish equivalent instrumento\textsubscript{B} para la estabilización\textsubscript{C} de las rentas\textsubscript{A} agrícolas\textsubscript{B}, to the French equivalent instrument\textsubscript{B} de stabilisation\textsubscript{C} du revenu\textsubscript{A} agricole\textsubscript{B}, to the Italian equivalent strumento\textsubscript{B} di stabilizzazione\textsubscript{C} dei redditi\textsubscript{A} agricoli\textsubscript{B}, to the Portuguese equivalent instrumento\textsubscript{B} de estabilização\textsubscript{C} do rendimento\textsubscript{A} agrícola\textsubscript{B} and to the Greek compound equivalent εργαλείο σταθεροποίησης γεωργικού εισοδήματος (lit: ‘instrument stabilization agricultural income’).

![Diagram of semantic equivalent trees for farm income stabilisation instrument](image)

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.
Table 25.5.: FTA for the semantically indeterminate farm income stabilisation instrument

<table>
<thead>
<tr>
<th>Rank</th>
<th>Bracketing</th>
<th>FTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[farm income] [stabilisation instrument]</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>[farm income] [stabilisation] instrument</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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\text{\textsubscript{A}} development\text{\textsubscript{B}} aid\text{\textsubscript{C}} projects\text{\textsubscript{D}}, which has five possible parse trees, shown in Figure 25.7.

When using the Italian phrase \textit{progetti\textsubscript{E} ecclesiastici\textsubscript{D} di aiuti\textsubscript{C} allo sviluppo\textsubscript{B}} (lit: ‘projects ecclesiastical of aid to development’), it is not possible to derive any valid parse tree, as illustrated with the Italien 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
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 \( \text{treeScore} \) given in Formula 25.7, where \( \text{freq}(st_i, STA) \) is the frequency of a subtree \( st_i \) in the STA, \( |L| \) is the number of support languages and \( \text{Cat}_\Delta \) is the \( \Delta \)-th Catalan number (Formula 25.5) with \( \Delta \) being the difference in the number of leaf nodes between \( ft \) and \( st_i \).
25. Compound Parsing Methods using Aligned Word Distance

\[
\text{treeScore}(ft) = \prod_{st_i \in ft} P(\text{valid} | st_i) \\
= \prod_{st_i \in ft} \frac{\text{freq}(\text{valid} \cap st_i)}{\text{freq}(st_i)} \\
= \prod_{st_i \in ft} \frac{\text{freq}(st_i, \text{STA})}{|L| \cdot \text{Cat} \Delta} \quad (25.7)
\]

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.

\[
P(\text{valid} | st_\alpha) = \frac{\text{freq}(st_\alpha, \text{STA})}{|L| \cdot \text{Cat} \Delta} \\
= \frac{4}{9 \cdot 2} = \frac{2}{9} \sim 0.222 \quad (25.8)
\]

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* development aid projects aligned to phrases of three support languages: Italian *progetti ecclesiastici di aiuti allo sviluppo* (lit: ‘projects ecclesiastical of aid to development’), German *kirchliche Entwicklungshilfeprojekte* ‘church {development aid projects}’ and French *projets d’ aide au développement* de l’ Église ‘projects of aid for development by the church’.

![Two possible parse trees for church development aid projects annotated with AWDs in Italian, German and French](image)

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-branching 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,...,j): the 4NC cannot be disambiguated between the distinctive structures i,...,j within the one-sentence context

FLAT: for expressions showing no internal structure, e.g., named entities like John A. Smith

SEMIND(i,...,j): the 4NC is semantically indeterminate, i.e., the structures i,...,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.

<table>
<thead>
<tr>
<th>Bracketing pattern</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td>A [B [C D]]</td>
<td>*</td>
</tr>
</tbody>
</table>
| 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.


\[
R\text{-Prec}(k\text{NC}) = \frac{|\text{top-}R(\text{sys trees}) \cap \text{gold trees}|}{|\text{top-}R(\text{sys trees})|}
\]

(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 \( \text{gold trees} \) 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 \( k\text{NC} \) 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 \( \text{MRP} \) takes the macro average of the \( R \)-Precision scores, as given in Formula 25.10

\[
\text{MRP}(\Omega) = \frac{\sum_{\Psi \in \Omega} \text{R-Prec}(\Psi)}{|\Omega|}
\]

(25.10)

where \( \Omega \) is the set of all target compounds \( \Psi \) (i.e., \( 3\text{NCs} \) and \( 4\text{NCs} \)) 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.

\[
P@t = \frac{|\text{top-}t(\text{sys trees}) \cap \text{gold trees}|}{|\text{top-}t(\text{sys trees})|}
\]

(25.11)

\[
R@t = \frac{|\text{top-}t(\text{sys trees}) \cap \text{gold trees}|}{|\text{gold trees}|}
\]

(25.12)

We present the macro average for \( P@t \) as \( \text{MP}@t \) and for \( R@t \) as \( \text{MR}@t \). Macro \( F_1 \) at \( t \) is the harmonic mean of \( \text{MP}@t \) and \( \text{MR}@t \). Since semantically indeterminate \( k\text{NCs} \) 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 \text{NFTAP} and \text{NSTAP} against \text{DBUP}. Since \text{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\textsubscript{rank}.

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 $1 \ldots 5$) 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\textsubscript{rank}, but the differences are small (average difference: 1.55%).

Considering the ratios of semantic indeterminacy, we can see that there are more
25. Compound Parsing Methods using Aligned Word Distance

<table>
<thead>
<tr>
<th>System</th>
<th>MRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFTAP</td>
<td>93.7%</td>
</tr>
<tr>
<td>NSTAP</td>
<td>94.0%</td>
</tr>
<tr>
<td>DBUP</td>
<td>92.6%</td>
</tr>
<tr>
<td>DBUP_rank</td>
<td>92.0%</td>
</tr>
<tr>
<td>FREQ</td>
<td>84.6%</td>
</tr>
<tr>
<td>CHANCE</td>
<td>62.5%</td>
</tr>
</tbody>
</table>

Table 25.7.: Parsing results in MRP for 3NCs and 4NCs

Semantic indeterminacy of 4NCs ($\frac{18}{33} = 54.5\%$) is higher 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.

<table>
<thead>
<tr>
<th>System</th>
<th>MRP</th>
<th>MP@1</th>
<th>MR@1</th>
<th>MF1@1</th>
<th>MP@2</th>
<th>MR@2</th>
<th>MF1@2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFTAP</td>
<td>70.0%</td>
<td>72.7%</td>
<td>47.5%</td>
<td>57.5%</td>
<td>60.6%</td>
<td>74.2%</td>
<td>66.7%</td>
</tr>
<tr>
<td>NSTAP</td>
<td>69.5%</td>
<td>69.7%</td>
<td>44.4%</td>
<td>54.2%</td>
<td>63.6%</td>
<td>78.8%</td>
<td>70.4%</td>
</tr>
<tr>
<td>DBUP</td>
<td>54.5%</td>
<td>69.7%</td>
<td>44.4%</td>
<td>54.2%</td>
<td>47.0%</td>
<td>59.1%</td>
<td>52.4%</td>
</tr>
<tr>
<td>DBUP_rank</td>
<td>62.9%</td>
<td>69.7%</td>
<td>44.4%</td>
<td>54.2%</td>
<td>54.5%</td>
<td>66.7%</td>
<td>60.0%</td>
</tr>
<tr>
<td>UPPER</td>
<td>86.0%</td>
<td>96.7%</td>
<td>67.2%</td>
<td>79.3%</td>
<td>70.0%</td>
<td>87.8%</td>
<td>77.9%</td>
</tr>
<tr>
<td>FREQ</td>
<td>60.1%</td>
<td>63.6%</td>
<td>38.4%</td>
<td>47.9%</td>
<td>56.1%</td>
<td>65.2%</td>
<td>60.3%</td>
</tr>
<tr>
<td>CHANCE</td>
<td>32.0%</td>
<td>39.4%</td>
<td>23.7%</td>
<td>29.6%</td>
<td>33.3%</td>
<td>42.4%</td>
<td>37.3%</td>
</tr>
</tbody>
</table>

Table 25.8.: Parsing results for 4NCs

Table 25.8 shows the results on cross-lingual compound parsing of 4NCs, where $\text{\ding{51}}$ means significantly outperformed by both NTAP methods; $\text{\ding{53}}$ means significantly outperformed by NFTAP or NSTAP. For the mean $R$-Precision, MRP, NFTAP and NSTAP significantly $^6$ outperform DBUP and DBUP\_rank. 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\_rank (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.

$^6$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 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 AWD 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 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\textsubscript{A} development\textsubscript{B} aid\textsubscript{C} projects\textsubscript{D} being aligned to the Italian phrasal equivalent progetti\textsubscript{B} ecclesiastic\textsubscript{A} di aiuti\textsubscript{C} allo sviluppo\textsubscript{D}. Due to the smallest AWD\textsubscript{it} between church\textsubscript{A} and projects\textsubscript{B}, 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): $SN_1 \text{ FC } SN_2 \text{ ADJ}$ (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 $SN_1 \text{ FC } SN_2$ 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 $SN \text{ ADJ } FC \text{ SN}$, 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 $risk_4 \text{ management}_4 \text{ decision}_4$ is aligned to the Swedish ternary closed compound $riskhanteringsbeslut_{ABC}$ and to the Portuguese phrase $decisão_{C} \text{ de gestão}_4 \text{ dos riscos}_4$. In both languages, $\text{AWD}(risk,management)$ is equal to $\text{AWD}(management,decision)$.

**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\textsubscript{token} outperforms APPP\textsubscript{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\textsubscript{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 Ngrams ($N > 1$) 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.
26. Bottom Line of the Compound Parsing

*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 Entwicklung* ‘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).

\[ RlogF(kNC_j) = \frac{F_j}{N_j} \cdot \log_2(F_j) \]  

(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-$l^1$ 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).

\[ AvgLog(APP_j) = \sum_{o=1}^{K_n} \frac{\log_2(F_o + 1)}{K_n} \]  

(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-

$^1$Thelen and Riloff (2002) used $l = 20 + i$ as number of collected items, which increases with the number of iterations $i$. 

370
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)| \tag{26.3}

The final formula for \(w_{AWD}\) 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.

\[
w_{AWD}(c_i, c_j) = \begin{cases} 
AWD(c_i, c_j) & \text{if } \Delta_{pos}(cAWp(c_i, c_j)) < 2 \\
1 + \sum_{k=cAWp(c_i, c_j)[1]}^{cAWp(c_i, c_j)[1]-1} w(\alpha_k) & \text{else}
\end{cases}
\tag{26.4}
\]

There are many possible weighting functions for \(w_{AWD}\). 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.

\[
w(\alpha) = \begin{cases} 
2 & \text{if } \text{contentword}(\alpha) \\
1 & \text{else}
\end{cases}
\tag{26.5}
\]

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.

\[
w(\alpha) = \frac{1}{freq(\alpha)}
\tag{26.6}
\]

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 ‘packages of television for cable’, 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 $wAWD$ 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 from the brain stem* pointing to a left-branched [[[brain stem] 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 $\Lambda$.

\[
WD_\Lambda(c_i,c_j) = |pos_{\Lambda}(c_i) - pos_{\Lambda}(c_j)|
\] (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\(^2\) title *Report on the Violations of Human Rights in the Conflict Zones* can provide the correct structure: $WD$(human, rights) = 1, $WD$(human, violations) = 2 and $WD$(rights, 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.

\(^2\)osce.org/odihr/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 development aid projects being aligned to the Italian phrasal equivalent progetti ecclesiastici di aiuti allo sviluppo (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 reserve currency being aligned to the French phrasal equivalent monnaie mondiale de reserve. 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.

\[
\begin{align*}
\text{world reserve currency} & \quad \text{world reserve currency} \\
\text{AWN}_f = 1 & \quad \text{AWN}_f = 1 \\
\text{world} & \quad \text{world} \\
\text{reserve} & \quad \text{reserve} \\
\text{currency} & \quad \text{currency} \\
\text{AWN}_f = 2 & \quad \text{AWN}_f = 0 \\
\text{AWN}_f = 0 & \quad \text{AWN}_f = 0 \\
\text{AWN}_f = 0 & \quad \text{AWN}_f = 0 \\
\text{AWN}_f = 3 & \quad \text{AWN}_f = 0 \\
\end{align*}
\]

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

<table>
<thead>
<tr>
<th>line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>if leaf node(N.RIGHT) then</td>
</tr>
<tr>
<td>2</td>
<td>return AWD(N.LEFT, N.RIGHT)</td>
</tr>
<tr>
<td>3</td>
<td>else</td>
</tr>
<tr>
<td>4</td>
<td>AWD_X ← AWD(N.LEFT, N.RIGHT)</td>
</tr>
<tr>
<td>5</td>
<td>AWD_Y ← AdjDepMod_AWD(N.LEFT, N.RIGHT.RIGHT)</td>
</tr>
<tr>
<td>6</td>
<td>if AWD_Y ≤ AWD_X then</td>
</tr>
<tr>
<td>7</td>
<td>N.dep ← TRUE</td>
</tr>
<tr>
<td>8</td>
<td>return AWD_Y</td>
</tr>
<tr>
<td>9</td>
<td>else</td>
</tr>
<tr>
<td>10</td>
<td>return AWD_X</td>
</tr>
<tr>
<td>11</td>
<td>end if</td>
</tr>
</tbody>
</table>

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.
twin pipe undersea gas pipeline

\[
\text{AdjDepMod}_\text{AWD} = ?
\]

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 \(AWD(\text{twin pipe, undersea gas pipeline})\), the root node is additionally annotated with \(\text{dep}\). The reason for marking the nodes with \(\text{dep}\) 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 \(\text{dep}\).

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 réserve, as shown in Figure 26.3.

For the best of our knowledge, this would be the first time, the dependency model is directly applied to a compound size \(k > 3\).
26.4. 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 \([A \mid B \ C]\), 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).

26.4.6. 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 counterintuitive, 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 \texttt{A B C} and a support language \texttt{l\textsubscript{sup}}. As input, a bilingual dictionary \texttt{dict\textsubscript{tgt\rightarrow l\textsubscript{sup}}} mapping from a target language \texttt{l\textsubscript{tgt}} to \texttt{l\textsubscript{sup}}, and a monolingual corpus in \texttt{l\textsubscript{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 \textbf{translated word distances} (TWDs) of all three \texttt{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 \texttt{numCommonUnits} is increased by the number of common units for the three current translations (line 9). For all three \texttt{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 \texttt{A} and \texttt{B}, the method returns \texttt{LEFT}, otherwise \texttt{RIGHT} (lines 21-24).

One of the main problems of this approach is that it accumulates \texttt{word} sense ambiguities from both the target language and the support language. For example, the English \texttt{word} \texttt{right} is listed with 24 translations in the \texttt{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 \texttt{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 $\text{DICT}_{\text{tgt} \rightarrow \text{lsup}}$

**Input 2:** Monolingual corpus context units for all support languages $\text{lsup}$

1: $\text{TWD}_A \_ B \leftarrow 0$
2: $\text{TWD}_A \_ C \leftarrow 0$
3: $\text{TWD}_B \_ C \leftarrow 0$
4: $\text{numCommonUnits} \leftarrow 0$
5: for translation $A_{lsup}$ in $\text{DICT}(A)$
6: for translation $B_{lsup}$ in $\text{DICT}(B)$
7: for translation $C_{lsup}$ in $\text{DICT}(C)$
8: $\text{commonUnits} = \{ u \mid u \text{ includes } A_{lsup}, B_{lsup} \text{ and } C_{lsup} \}$
9: $\text{numCommonUnits} \leftarrow \text{numCommonUnits} + |\text{commonUnits}|$
10: for $u \in \text{commonUnits}$
11: $\text{TWD}_A \_ B \leftarrow \text{TWD}_A \_ B + |\text{pos}(\text{DICT}(A)) - \text{pos}(\text{DICT}(B))|$
12: $\text{TWD}_A \_ C \leftarrow \text{TWD}_A \_ C + |\text{pos}(\text{DICT}(A)) - \text{pos}(\text{DICT}(C))|$
13: $\text{TWD}_B \_ C \leftarrow \text{TWD}_B \_ C + |\text{pos}(\text{DICT}(B)) - \text{pos}(\text{DICT}(C))|$
14: end for
15: end for
16: end for
17: $\text{TWD}_A \_ B \leftarrow \text{TWD}_A \_ B / \text{numCommonUnits}$
18: $\text{TWD}_A \_ C \leftarrow \text{TWD}_A \_ C / \text{numCommonUnits}$
19: $\text{TWD}_B \_ C \leftarrow \text{TWD}_B \_ C / \text{numCommonUnits}$
20: if $\text{TWD}_A \_ B$ is smallest then
21: return LEFT
22: else
23: return RIGHT
24: 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.

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)
27. Summary, Conclusion and Future Work of the Thesis

(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 ($\rightarrow$ Inseparability)
2. the non-final constituents of $\Psi$ cannot be modified by external words ($\rightarrow$ Inability to modify the modifier)
3. one of the non-final constituents of $\Psi$ gets a prosodic stress ($\rightarrow$ Prosody)
4. there are some compound equivalents of $\Psi$, as given in a parallel corpus ($\rightarrow$ 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).
27. Summary, Conclusion and Future Work of the Thesis

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 $\Xi_{\text{closed}}$ 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.

![Compound Analysis in NLP](image)

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.

![Ambiguity in various Compound Analysis Levels](image)

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\(^1\) (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.

\(^1\)ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html
A. Universal Surface Patterns

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 conlates 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.

<table>
<thead>
<tr>
<th>Word seq $\omega$</th>
<th>PoS pattern</th>
<th>USP $\Lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mann (man)</td>
<td>NN</td>
<td>SN</td>
</tr>
<tr>
<td>dood</td>
<td>straf (death penalty)</td>
<td>NN</td>
</tr>
<tr>
<td>van de (of the)</td>
<td>prep det_art</td>
<td>FC</td>
</tr>
<tr>
<td>einkommensteuerrechtlichen ({income tax})$_{ADJ}$</td>
<td>ADJ</td>
<td>3ADJC</td>
</tr>
<tr>
<td>rad</td>
<td>fahren (to cycle)</td>
<td>VVINF</td>
</tr>
<tr>
<td>Kernarbeitsnormen (core labour standards)</td>
<td>NN</td>
<td>3NC</td>
</tr>
<tr>
<td>Rechts</td>
<td>vorschriften für die Wettbewerbs</td>
<td>politik (legislations for competition policy )</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>Example</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>null operation</td>
<td>Kohlsuppe ‘cabbage soup’</td>
<td>22,759</td>
</tr>
<tr>
<td>+s</td>
<td>s-suffix</td>
<td>Staatsfeind ‘public enemy’</td>
<td>9637</td>
</tr>
<tr>
<td>+n</td>
<td>n-suffix</td>
<td>Soziologenkongress ‘sociologist congress’</td>
<td>5307</td>
</tr>
<tr>
<td>+en</td>
<td>en-suffix</td>
<td>Straußeni ‘ostrich egg’</td>
<td>4316</td>
</tr>
<tr>
<td>+nen</td>
<td>nen-suffix</td>
<td>Wöchnerinnenheim ‘maternity home’</td>
<td>2610</td>
</tr>
<tr>
<td>+us, +en</td>
<td>us-trunc., en-suffix</td>
<td>Aphorismenschatz ‘aphorism lexicon’</td>
<td>618</td>
</tr>
<tr>
<td>+um, +en</td>
<td>um-trunc., en-suffix</td>
<td>Museenverwaltung ‘museum administration’</td>
<td>348</td>
</tr>
<tr>
<td>+um, +a</td>
<td>um-trunc., a-suffix</td>
<td>Aphrodisiaka</td>
<td>verkäufer ‘aphrodisiac seller’</td>
</tr>
<tr>
<td>-e</td>
<td>e-trunc.</td>
<td>Kirchhof ‘churchyard’</td>
<td>122</td>
</tr>
<tr>
<td>-a, -en</td>
<td>a-trunc., en-suffix</td>
<td>Madonna</td>
<td>kult ‘Madonna worship’</td>
</tr>
<tr>
<td>-er</td>
<td>e-suffix</td>
<td>Hunde</td>
<td>halter ‘dog owner’</td>
</tr>
<tr>
<td>-en</td>
<td>en-trunc.</td>
<td>Gänsgelklein ‘goose giblets’</td>
<td>73</td>
</tr>
<tr>
<td>+en</td>
<td>en-trunc.</td>
<td>Stadien</td>
<td>verbot ‘stadium ban’</td>
</tr>
<tr>
<td>+es</td>
<td>es-suffix</td>
<td>Geistes</td>
<td>haltung ‘attitude’</td>
</tr>
<tr>
<td>-er</td>
<td>e-trunc.</td>
<td>Blätter</td>
<td>walde ‘leaf forest’</td>
</tr>
<tr>
<td>-en</td>
<td>en-trunc.</td>
<td>Südwind ‘south wind’</td>
<td>33</td>
</tr>
<tr>
<td>+en, +a</td>
<td>on-trunc., a-suffix</td>
<td>Pharmaka</td>
<td>analyse ‘pharmaceutical analysis’</td>
</tr>
<tr>
<td>-er</td>
<td>er-suffix</td>
<td>Geister</td>
<td>stunde ‘witching hour’</td>
</tr>
<tr>
<td>-ien</td>
<td>ien-suffix</td>
<td>Prinzipi</td>
<td>reiter ‘stickler for principles’</td>
</tr>
<tr>
<td>-e, +i</td>
<td>e-trunc., i-suffix</td>
<td>Carabinier</td>
<td>schule ‘carabinier school’</td>
</tr>
</tbody>
</table>

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:** \( \text{lemmas} \) \{list of constituent lemmas, resulted from splitting \( c \)\}

1. \( \text{forms} \leftarrow [ \ ] \) \{the final constituent forms for the SPF\}
2. \( \text{stem} \leftarrow c \) \{the stem is initialized with the full compound \( c \)\}
3. while \( |\text{lemmas}| > 1 \) do
4. \( \text{lastLemma} = \text{lastElement}(\text{lemmas}) \)
5. if \( \text{stem}.\text{endsWith}(\text{lastLemma}) \) then
6. \( \text{forms} \leftarrow \text{lastLemma} + \text{forms} \) \{last lemma is prepended to forms\}
7. \( \text{stem} \leftarrow \text{stem} - \text{lastLemma} \) \{last lemma is truncated from the stem\}
8. else
9. for suffix of stem do
10. \( \text{score(suffix)} = \text{len(suffix)} / \text{ED(lastLemma, suffix)} \)
11. end for
12. \( \text{inflForm} = \text{suffix(maxScore)} \) \{get suffix with maximum score\}
13. \( \text{forms} \leftarrow \text{inflForm} + \text{forms} \) \{inflected form is prepended to forms\}
14. \( \text{stem} \leftarrow \text{stem} - \text{inflForm} \) \{inflected form is truncated from the stem\}
15. end if
16. \( \text{lemmas} \leftarrow \text{lemmas} - \text{lastLemma} \) \{last lemma is removed from the lemmas\}
17. end while
18. \( \text{forms} \leftarrow \text{stem} + \text{forms} \) \{stem as first constituent form\}
19. return \( \text{join}(|,\text{forms}) \) \{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

\[ c_{\text{form}} = \text{Supphähnnerzucht} \]
\[ c_{\text{lem}} = \text{Supphähnnerzucht} \]

\[
\begin{array}{c}
\text{1} \\
\text{2} \\
\text{3} \\
\text{4} \\
\text{5}
\end{array}
\]

\[ c_{\text{form}} = \text{Supphähnner} \]
\[ c_{\text{lem}} = \text{Supphähnner} \]
\[ c_{\text{form}} = \text{Zucht} \]
\[ c_{\text{lem}} = \text{Zucht} \]
\[ c_{\text{form}} = \text{Suppen} \]
\[ c_{\text{lem}} = \text{Suppen} \]
\[ c_{\text{form}} = \text{Huhn} \]
\[ c_{\text{lem}} = \text{Huhn} \]

Figure C.1.: Example of a split tree with recursive lemma splitting

The main issue is based on the fact that \( c_{\text{form}} = c_{\text{lem}} \) for the node \( \circ \). So, how can we propagate the split point in \( \text{Suppen | huhn} \) to the constituent form \( \text{Supphähnner} \) (for arriving at the SPF \( \text{Suppen | hühner} \))? 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.

\[
SPF(N) = \begin{cases} 
  c_{\text{form}}(N) & \text{leaf}(N) \\
  SPF(N.LD) | SPF(N.RD)[head/MOP_{c_{\text{lem}}(N)\rightarrow c_{\text{form}}(N)}(head)] & \text{else}
\end{cases} 
\tag{C.1}
\]

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 \( c_{\text{lem}} \) of \( N \) to the \( c_{\text{form}} \) of \( N \).

For example, in Figure C.1, node \( \circ \), the MOP for transforming \( \text{Supphähnner} \) to \( \text{Supphähnner} \), is \( u/ü:\$/$er\$. The head of the SPF in node \( \circ \) is the constituent form, \( \text{huhn} \). As a result, the SPF of node \( \circ \) is ‘\( SPF(\circ) \mid SPF(\circ)[\text{huhn/hühner}] \)’ = \( \text{Suppen | hühner} \).
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 \textit{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, \textit{Suppenhühnerzucht}, and the list of mother-node related \textit{cforms}: \{\textit{Suppen}, \textit{Huhn}, \textit{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.

\textbf{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.

<table>
<thead>
<tr>
<th>Gold standard</th>
<th>Size</th>
<th># SPF mismatches</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HH2011GS</td>
<td>54,148</td>
<td>8</td>
<td>99.9852%</td>
</tr>
<tr>
<td>M2006GS</td>
<td>139,081</td>
<td>15</td>
<td>99.9892%</td>
</tr>
<tr>
<td>HB2008GS</td>
<td>687</td>
<td>0</td>
<td>100%</td>
</tr>
</tbody>
</table>

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 SPF provided
by FF2010 are not correct. For example, while the linear SPF compiler produces \textit{wett|tauchen} for the compound \textit{Wetttauchen} ‘diving race’ and the constituent lemmas \textit{wetten} ‘to bet’ and \textit{tauchen} ‘to dive’, FF2010 provides the incorrect SPF \textit{wetten|tauchen}. Actually, when discounting wrong SPF mismatches, the linear SPF compilation method even achieves accuracies of 99.9945\% (HH2011GS) and 99.995\% (M2006GS).

\textbf{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.

\textbf{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.

\textbf{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
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 *Kuckucksei*er ‘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 *Kuckucksei*er 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, SP\(P\), 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 SPF 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.


Cap (2014) developed a compound splitting gold standard (C2014GS) derived from the development set of the 2007 Workshop on Statistical Machine Translation\(^1\). 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 *Untersuchungshäftling*, 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’.

\(^1\)http://www.statmt.org/wmt07/shared-task.html
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 $G_{\text{bst-NN}}$, 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.

<table>
<thead>
<tr>
<th></th>
<th>HH2011GS</th>
<th>M2006GS</th>
<th>HB2008GS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{\text{bst-NN}}$</td>
<td>743</td>
<td>125</td>
<td>——</td>
</tr>
<tr>
<td>$G_{\text{bst-NN}}$</td>
<td>176,619</td>
<td>——</td>
<td>——</td>
</tr>
</tbody>
</table>

Table D.1.: Splitting gold standard overlap between $HH2011GS + M2006GS + HB2008GS$ and $\text{Ghost-NN}$

Table D.1 shows the overlap of $G_{\text{bst-NN}}$ with the union of $HH2011GS$, $M2006GS$ and $HB2008GS$. All four gold standards have 743 compounds in common and there are only 125 samples in $G_{\text{bst-NN}}$, which are not covered by the three gold standards used in our experiments. This which means that the majority of $G_{\text{bst-NN}}$ 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. Annotation Guidelines for Creating the Europarl Nominal Compoundhood Ratings

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*).

\[ \Rightarrow \text{Does the spelling of the expression under consideration (i.e., closed or open compounding) point to compoundhood?} \]

[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.

\[ \Rightarrow \text{Can you think of a way to insert an element between the constituents of the underlying expression?} \]

[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*)
E. Annotation Guidelines for Creating the Europarl Nominal Compoundhood Ratings

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?
E. Annotation Guidelines for Creating the Europarl Nominal Compoundhood Ratings

[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. Annotation Guidelines for Creating the Europarl Nominal Compoundhood Ratings

### 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.

```plaintext
<table>
<thead>
<tr>
<th>ID</th>
<th>SENTENCE</th>
<th>COMPOUND</th>
<th>COMPOUND RATING</th>
<th>SPACING</th>
<th>INSEPARABILITY</th>
<th>ONE-REPLACEMENT</th>
<th>REPLACEMENT</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>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</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>at the same time, we should consider introducing cultural clubs for retirees and for all those working in the cultural sector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>the Commission will bear the issues in question in mind within the framework of the development of our policy in this sector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>the majority of this Parliament...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>the Commission invited to reinforce in order to improve the effectiveness of the Euro-Mediterranean partnership</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>In present, providing social and economic development in these states is undeniably the imperative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>Here very valid questions have been asked</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>a member of the South Asian Association for Regional Cooperation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>It has visited Bangladesh several times</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Finally, clear political support must be given to every country that takes a leadership role, whatever it may be, which is not the case in practice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**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.
4. 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)*

5. Afterwards, annotate all subsequent columns (columns E to J) with the ratings for the linguistic criteria described in Section E.2.

6. 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.

   ![Figure E.2.: Examples of spreadsheet-based nominal compound annotation](image)

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

4NC

5NC
five-noun Compound. 319, 355

A

AdjDepMod

Adjacency-Dependency Model. 299, 355, 356, 390
List of Abbreviations

AdjMod

AM

APP

APPP

APPPA
aligned phrase pattern parsing supported by word alignment. 285, 311, 313, 315, 344, 397

AWD

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

414
List of Abbreviations

CCR
Closed Compound Restrictor. 140–144, 147–149, 151, 153, 154, 312, 320, 338, 364, 395

CRF
Conditional Random Fields. 89

D

DBUP

DC
Determination of Compoundhood. 4

DepMod

DS

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
List of Abbreviations

ENCD

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

416
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

M

ME
Maximum Entropy. 173, 299, 303

MI
Mutual Information. 91, 92, 166, 172, 287, 288, 291

MOP

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

417
List of Abbreviations

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

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
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOV</td>
<td>Out-Of-Vocabulary.</td>
<td>167, 177</td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCFG</td>
<td>Probabilistic Context-Free Grammar.</td>
<td>303, 304</td>
</tr>
<tr>
<td>PMI</td>
<td>Pointwise Mutual Information.</td>
<td>92, 94, 294, 300–302</td>
</tr>
<tr>
<td>PP</td>
<td>Prepositional Phrase.</td>
<td>10, 34, 43, 44, 91, 288, 290, 303, 343</td>
</tr>
<tr>
<td>PTB</td>
<td>Penn Treebank.</td>
<td>276, 297–300</td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QA</td>
<td>Question Answering.</td>
<td>11, 240, 250, 273</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHHR</td>
<td>RightHand Head Rule.</td>
<td>33, 189</td>
</tr>
</tbody>
</table>
List of Abbreviations

RTE

S
SA
structural analysis. 5

SemRel
semantic relation. 293

SiMode
Similarity Mode. 228, 229, 232, 234–238, 258, 263, 266, 267, 271

SMT

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

W

wFST
weighted Finite State Transducer. 172, 174

WSD
Word Sense Disambiguation. 8, 367, 379

X

XCI
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 Σ ........ 370
26.2 AdjDepMod Annotation Function ............................ 375
26.3 Non-parallel approach to cross-lingually parsing a compound A B C .......... 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 Hühnersuppenrezept ............................. 6
1.4 Split tree for Hühnersuppenrezept .................................... 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 Hochschulgebäude ‘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 *Quartal* ‘quarter (year)’ ............................. 207
18.5 False compound split of *Läuferteam* ‘runner’s team’ ......................... 209
18.6 Linguistically motivated split of *Schauspielzeitschrift* ....................... 210
18.7 Tree pruning for *Schauspielzeitschrift* ........................................... 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 *Langsamfahrstelle* .......... 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 *energy efficiency action plans* ................. 296

23.1 Dependency relations for LEFT- and RIGHT-branched *natural language
    processing* ....................................................................................... 302
23.2 Dependency relations for swapped modifiers in a RIGHT-branched TC . 302
23.3 Parse tree for *wooden French onion soup bowl handle* ....................... 315

25.1 DBUP parse tree for *air transport safety organization* ....................... 339
25.2 DBUP parse tree for *twin pipe undersea gas pipeline* ....................... 340
25.3 Partial result for parsing *book price fixing schemes* using a Danish phrase 346
25.4 Possible parse trees for *air traffic control* ....................................... 347
25.5 Possible binary parse trees for *energy efficiency action plan* .............. 351
25.6 Semantically equivalent trees for *farm income stabilisation instrument* 352
25.7 Possible binary parse trees for *church development aid projects* .......... 354
25.8 Two possible parse trees for *church development aid projects* annotated
    with AWDs in Italian, German and French ......................................... 357

26.1 Trees for *world reserve currency* .................................................... 374
26.2 Tree structure for *twin pipe undersea gas pipeline* ............................ 376
26.3 AdjDepMod-annotated parse trees for *world reserve currency* ............ 376

27.1 Thematic structure of the thesis ......................................................... 383
27.2 Compound Analysis in NLP ............................................................... 387

426
# 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


2015


2016

List of Figures

2017

⇒ presented and elaborated in Part D, Chapter 20
# List of Tables

3.1 Recursive compound construction for *orange juice* .......................... 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 *made_of* 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

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>EUROPARL language selection</td>
<td>120</td>
</tr>
<tr>
<td>10.1</td>
<td>Size of the final ENCR datasets</td>
<td>127</td>
</tr>
<tr>
<td>10.2</td>
<td>Agreement on compound extraction in the <em>training stage</em></td>
<td>129</td>
</tr>
<tr>
<td>10.3</td>
<td>Agreement on compound extraction in the <em>agreement set</em></td>
<td>129</td>
</tr>
<tr>
<td>10.4</td>
<td>Exclusive extractions for both annotators</td>
<td>130</td>
</tr>
<tr>
<td>10.5</td>
<td>Average difference in compoundhood rating in the <em>training stage</em></td>
<td>130</td>
</tr>
<tr>
<td>10.6</td>
<td>Average difference in compoundhood rating in the <em>agreement set</em></td>
<td>131</td>
</tr>
<tr>
<td>10.7</td>
<td>Average difference in LC rating in the <em>training stage</em> (1)</td>
<td>131</td>
</tr>
<tr>
<td>10.8</td>
<td>Agreement on LC rating in the <em>training stage</em> (2)</td>
<td>132</td>
</tr>
<tr>
<td>10.9</td>
<td>Agreement on LC rating in the <em>agreement set</em></td>
<td>133</td>
</tr>
<tr>
<td>10.10</td>
<td>Compoundhood prediction accuracy using a J48 decision tree</td>
<td>135</td>
</tr>
<tr>
<td>10.11</td>
<td>Compoundhood prediction using feature groups in a J48 decision tree</td>
<td>136</td>
</tr>
<tr>
<td>10.12</td>
<td>Statistical significance test for a J48 decision tree</td>
<td>136</td>
</tr>
<tr>
<td>10.13</td>
<td>Compoundhood prediction accuracy using a Naive Bayes classifier</td>
<td>138</td>
</tr>
<tr>
<td>10.14</td>
<td>Compoundhood prediction using feature groups in a Naive Bayes classifier</td>
<td>138</td>
</tr>
<tr>
<td>10.15</td>
<td>Statistical significance test for a Naive Bayes Classification</td>
<td>138</td>
</tr>
<tr>
<td>10.16</td>
<td>Average difference between compoundhood and linguistic criteria</td>
<td>139</td>
</tr>
<tr>
<td>10.17</td>
<td>XCI - USP frequency distribution for all aligned languages</td>
<td>143</td>
</tr>
<tr>
<td>10.18</td>
<td>XCI - USP frequency distribution for Germanic languages</td>
<td>143</td>
</tr>
<tr>
<td>10.19</td>
<td>XCI - USP frequency distribution for Greek</td>
<td>144</td>
</tr>
<tr>
<td>10.20</td>
<td>XCI - USP frequency distribution for Romance languages</td>
<td>144</td>
</tr>
<tr>
<td>11.1</td>
<td>Predefined PoS patterns for the compound candidate selection</td>
<td>148</td>
</tr>
<tr>
<td>12.1</td>
<td>PoS pattern distribution in CCR(<em>n</em>)</td>
<td>154</td>
</tr>
<tr>
<td>12.2</td>
<td>The 10 most frequent English JJ NN sequences in CCR(0) and CCR(4)</td>
<td>155</td>
</tr>
<tr>
<td>12.3</td>
<td>The 10 most frequent English NN POS NN sequences in CCR(0) and CCR(4)</td>
<td>155</td>
</tr>
<tr>
<td>12.4</td>
<td>Degree of closed compounding among the closed compounding languages</td>
<td>156</td>
</tr>
<tr>
<td></td>
<td>in CCR(1)</td>
<td>156</td>
</tr>
<tr>
<td>12.5</td>
<td>The 10 most frequent equivalents of English 3NCs in USP format</td>
<td>157</td>
</tr>
<tr>
<td>13.1</td>
<td>Compound Identification Results for <em>Annotation1</em> dataset</td>
<td>160</td>
</tr>
<tr>
<td>13.2</td>
<td>Compound Identification Results for <em>Annotation2</em> dataset</td>
<td>161</td>
</tr>
<tr>
<td>13.3</td>
<td>Compound Identification Results for <em>Combination</em> dataset</td>
<td>161</td>
</tr>
<tr>
<td>16.1</td>
<td>Splitting example from Larson et al. (2000)</td>
<td>182</td>
</tr>
</tbody>
</table>
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 German, Dutch and Afrikaans  196

18.1 Word MOPs for the lemma Termin ‘appointment’  206
18.2 Word MOPs for the lemma Lauf ‘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 comparison; 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 comparison  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 APPPWA ................................................................. 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 farm income stabilisation instrument ........ 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

434
## 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
List of Terms

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 discontiguous) 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., $\text{data}_1 \text{ base}_2$ or $\text{blue}_1$-$\text{eyed}_2$). 17, 18, 29, 46, 48, 51, 55, 61, 68, 82, 99, 101, 103, 105, 106, 137, 184–186, 189–192,
List of Terms


**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**


**closed compounding**


**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

437
**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**


**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
**List of Terms**

**compound equivalent**

A cross-lingual equivalent being a **compound**, e.g., a cross-lingually aligned closed compound. 151, 332, 363, 365, 405

**compound parsing**


**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**


**compoundhood**


**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**


439
**List of Terms**

**constituent**


**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 discontiguous 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**


**constituent inflection**


440
List of Terms

constituent lemma


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, \texttt{draft}_1 \texttt{treaty}_2 translated to German as \textit{Verfassungs}$_2$\textit{entwurf}$_1$. 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., \textit{noun}, \textit{verb}, \textit{adjective}, \textit{proper name}, \textit{adverb}, \ldots. 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 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
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

dendocentric compound

“An 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 *witsneus* ‘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 *witsneus* is *bahuvrihi compound.*” (Online Lexicon of Linguistics\(^1\)) . 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 \(\mid\) otter* ‘oath otter’ rather than into *Ei \(\mid\) dotter* ‘egg yolk’, or the correctly identified *constituent forms* are falsely

\(^1\)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**

List of Terms

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

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 α directly
List of Terms

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


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

446
**List of Terms**

**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**


**MOP application**

The application of a Morphological Operation Pattern (MOP) on a string Σ resulting in a string Ω. 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
List of Terms

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


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


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

448
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 μ, 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
List of Terms

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


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
List of Terms

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 Ngrams, 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
List of Terms

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
List of Terms


synthetic compound

A compound with a deverbal head (e.g., truck driver). 9, 10, 42, 47, 52, 153

T

target compound


target constituent


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$_1$ rights$_2$ abuse$_3$). 29, 44, 89, 137, 269, 275, 283, 309, 312, 343, 346, 388, 398, 405, 409
List of Terms

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 $+$ in the German Armuts ‘poverty+s’ as in Armutsbekämpfung ‘poverty elimination’. 49

V
List of Terms

verb compound
A compound composed of only verbs (e.g., to freeze-dry but not to fingerprint).

verbal compound
A compound with a verbal head (e.g., to fingerprint).

W

word
A word is an isolated unit, occurring typically separated by whitespace or punctuation characters (e.g., a closed compound).

word alignment
The (commonly automatic) process (or result) of mapping words from one language to their equivalents in another language, in a parallel corpus.

word alignment error
False mappings in the word alignment (e.g., missing or spurious links).

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.

word form
A word form is a possibly word-inflected embodiment of a lexeme in context.
**word inflection**


**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


Church, K. and Patil, R. (1982). Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table. *Computational Linguistics*.


Iordachioaia, G. (2017). Deverbal Compounds between Nominalizations and Phrasal


BIBLIOGRAPHY

Association for Computational Linguistics.
BIBLIOGRAPHY

homes: How compositional distributional semantics can help NP parsing. In In Proceedings of EMNLP.


BIBLIOGRAPHY


BIBLIOGRAPHY

Springer Berlin Heidelberg, Berlin, Heidelberg.


473
and the Shared Task: Semantic Textual Similarity, pages 266–275, Atlanta, Georgia, USA. Association for Computational Linguistics.


BIBLIOGRAPHY

Springer Berlin Heidelberg, Berlin, Heidelberg.


Weller, M., Cap, F., Müller, S., Schulte im Walde, S., and Fraser, A. (2014). Distin-


Ziering, P. and Van der Plas, L. (2014). What good are ‘Nominalkomposita’ for ‘noun


