Morphological Processing of Compounds for Statistical Machine Translation

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

Vorgelegt von

Fabienne Cap
aus Donaueschingen

Hauptberichter:	Prof. Dr. Jonas Kuhn
1. Mitberichter:	Dr. Jörg Tiedemann
2. Mitberichter:	Dr. Alexander M. Fraser

Tag der mündlichen Prüfung: 17. Oktober 2014

Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

2014

Finding a paradise wasn't easy, but still,
there's a road going down the other side of this hill.

(G.Barlow)

Publications

Parts of the research described in this thesis have been published in:

Cap et al. (2014a) Fabienne Cap, Alexander Fraser, Marion Weller and Aoife Cahill (2014) *How to Produce Unseen Teddy Bears – Improved Morphological Processing of Compounds in SMT.* In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL) 2014, Göteborg/Sverige, pp. 579–587.

Cap et al. (2014b) Fabienne Cap, Marion Weller, Anita Ramm and Alexander Fraser (2014) *CimS – The CIS and IMS Joint Submission to WMT 2014 Translating from English into German.* In: Proceedings of the 9th Workshop on Statistical Machine Translation (WMT) 2014, Translation task submissions; Baltimore/USA, p. 71–78.

Fraser et al. (2012) Alexander Fraser, Marion Weller, Aoife Cahill and Fabienne Cap (2012) *Modeling Inflection and Word Formation in SMT.* In: Proceedings of the 13th Conference of the Euorpean Chapter of the Association for Computational Linguistics (EACL) 2012, Avignon/France, pp. 664–674.

Fritzinger and Fraser (2010) Fabienne Fritzinger and Alexander Fraser (2010) *How to Avoid Burning Ducks – Combining Linguistic Analysis and Corpus Statistics for German Compound Processing.* In: Proceedings of the 5th Workshop on Statistical Machine Translation (WMT) 2010, Uppsala/Sverige, pp. 224–234.

Abstract

Machine Translation denotes the translation of a text written in one language into another language performed by a computer program. In times of internet and globalisation, there has been a constantly growing need for machine translation. For example, think of the European Union, with its 24 official languages into which each official document must be translated. The translation of official documents would be less manageable and much less affordable without computer-aided translation systems.

Most state-of-the-art machine translation systems are based on statistical models. These are trained on a bilingual text collection to "learn" translational correspondences of words (and phrases) of the two languages. The underlying text collection must be parallel, i.e. the content of one line must exactly correspond to the translation of this line in the other language. After training the statistical models, they can be used to translate new texts. However, one of the drawbacks of *Statistical Machine Translation* (SMT) is that it can only translate words which have occurred in the training texts.

This applies in particular to SMT systems which have been designed for translating from and to German. It is widely known that German allows for productive word formation processes. Speakers of German can put together existing words to form new words, called *compounds*. An example is the German *"Apfel + Baum = Apfelbaum"* (= "apple + tree = apple tree"). Theoretically there is no limit to the length of a German compound. Whereas *"Apfelbaum"* (= "apple tree") is a rather common German compound, *"Apfelbaumholzpalettenabtransport"* (= "apple|tree|wood|pallet|removal") is a spontaneous new creation, which (probably) has not occurred in any text collection yet. The productivity of German compounds leads to a large number of distinct compound types, many of which occur only with low frequency in a text collection, if they occur at all. This fact makes German compounds a challenge for SMT systems, as only words which have occurred in the parallel training data can later be translated by the systems. Splitting compounds into their component words can solve this problem. For example, splitting *"Apfelbaumholzpalettenabtransport"* into its component words, it becomes intuitively clear that *"Apfel"* (= "apple"), *"Baum"* (= "tree"), *"Palette"* (= "palette") and *"Abtransport"* (= "removal") are all common German words, which should have occurred much more often in any text collection than the compound as a whole. Splitting compounds thus potentially makes them translatable part-by-part.

This thesis deals with the question as to whether using morphologically aware compound splitting improves translation performance, when compared to previous approaches to compound splitting for SMT. To do so, we investigate both translation directions of the language pair German and English. In the past, there have been several approaches to compound splitting for SMT systems for translating from German to English. However, the problem has mostly been ignored for the opposite translation direction, from English to German. Note that this translation direction is the more challenging one: prior to training and translation, compounds must be split and after translation, they must be accurately reassembled. Moreover, German has a rich inflectional morphology. For example, it requires the agreement of all noun phrase components which are morphologically marked. In this thesis, we introduce a compound processing procedure for SMT which is able to put together new compounds that have not occurred in the parallel training data and inflects these compounds correctly – in accordance to their context. Our work is the first which takes syntactic information, derived from the source language sentence (here: English) into consideration for our decision which simple words to merge into compounds.

We evaluate the quality of our morphological compound splitting approach using manual evaluations. We measure the impact of our compound processing approach on the translation performance of a state-of-the-art, freely available SMT system. We investigate both translation directions of the language pair German and English. Whenever possible, we compare our results to previous approaches to compound processing, most of which work without morphological knowledge.

Deutsche Zusammenfassung

Der Begriff *Maschinelle Übersetzung* beschreibt Übersetzungen von einer natürlichen Sprache in eine andere unter Zuhilfenahme eines Computers oder Computerprogramms. In Zeiten des Internets und zunehmender Globalisierung sind maschinelle Übersetzungssysteme allgegenwärtig geworden. Man denke nur an die Europäische Union, mit ihren 24 offiziellen Amtssprachen, in welchen jedes offizielle EU-Dokument vorliegen muss. Die Übersetzungen offizieller Dokumente wären ohne computer-gestützte Systeme kaum zu bewältigen, vor allem aber wären sie unbezahlbar.

Heutige maschinelle Übersetzungssysteme basieren zumeist auf statistischen Modellen. Diese werden auf einer zweisprachigen Textmenge trainiert um Wortentsprechungen beider Sprachen zu "lernen". Die zugrundeliegende Textmenge, bestehend aus Millionen von Sätzen, muss in paralleler Form vorliegen, d.h. der Inhalt jeder Zeile muss genau der Übersetzung dieser Zeile in der anderen Sprache entsprechen. Nachdem die statistischen Modelle trainiert wurden, können sie dann auf die Übersetzung von neuen Texten angewandt werden. Ein entscheidender Nachteil der *Statistischen Maschinellen Übersetzung* (SMÜ) ist, dass nur Wörter und Konstrukte übersetzt werden können, die zuvor in der großen Trainingstextmenge vorgekommen sind.

Dies gilt insbesondere für SMÜ Systeme, die für die Übersetzung von und nach Deutsch konzipiert sind. Die deutsche Sprache ist weitgehend bekannt für ihre produktiven Wortbildungsprozesse. Sprecher des Deutschen können jederzeit durch Zusammensetzung bereits vorhandener Wörter neue Wörter bilden, sogenannte *Komposita*. Ein Beispiel hierfür ist "Apfel+Baum = Apfelbaum". Deutsche Komposita können theoretisch unendlich lang werden. Wohingegen "Apfelbaum" ein recht gebräuchliches und dadurch häufig vorkommendes Kompositum ist, ist "Apfelbaumholzpalettenabtransport" eine spontane Neubildung, für die es (vermutlich) noch keine Belege gibt. Durch die Produktivität deutscher Komposita, kommt es zu einer sehr hohen Anzahl an verschiedenen Komposita-Typen, von denen wiederum viele nur selten (oder auch gar nicht) in Texten vorgekommen sind. Diese Tatsache macht deutsche Komposita problematisch für SMÜ

Systeme, da nur Wörter, die in den Trainingstexten vorgekommen sind, auch von den Systemen übersetzt werden können. Die Zerlegung von Komposita in ihre Einzelwörter kann hierbei Abhilfe schaffen. Wenn man z.B. “Apfelbaumholzpalettenabtransport” in seine Bestandteile zerlegt, wird schnell klar, daß “Apfel”, “Baum”, “Holz”, “Palette,” und “Abtransport” alles gewöhnliche deutsche Wörter sind, die eher in den Trainingstexten vorgekommen sind als das Kompositum an sich. Die Zerlegung von Komposita macht sie also potentiell Wort für Wort übersetzbar.

Diese Dissertation befasst sich mit der Frage ob durch Zerlegung deutscher Komposita mithilfe morphologischen Wissens die Übersetzungsqualität eines SMÜ Systems verbessert werden kann, im Vergleich zu früheren Methoden zur Kompositazerlegung. Wir untersuchen hierfür beide Übersetzungsrichtungen des Sprachpaares Deutsch und Englisch. Wohingegen es schon einige verfügbare Ansätze zur Kompositazerlegung für SMÜ Systeme von Deutsch nach Englisch gibt, ist das Problem für die entgegengesetzte Übersetzungsrichtung von Englisch nach Deutsch bisher weitgehend ignoriert worden. Man bedenke zum einen, dass bei einer Übersetzung vom Englischen ins Deutsche die deutschen Komposita nicht nur vor der Übersetzung zerlegt werden müssen, sondern sie müssen auch anschließend wieder korrekt zusammengefügt werden. Zum anderen verfügt das Deutsche über eine reiche Flexionsmorphologie, die z.B. die Übereinstimmung aller morphologisch markierten Merkmale innerhalb einer Nominalphrase verlangt. Wir stellen in dieser Dissertation erstmals ein Werkzeug zur Kompositabehandlung in SMÜ vor, das bei Bedarf Komposita zusammenfügen kann, die in den Trainingstexten nicht vorgekommen sind und außerdem diese Komposita – in Abhängigkeit ihres unmittelbaren Kontextes – mit einer korrekten Flexionsendung versehen kann. Die Entscheidung darüber, welche Einzelwörter nach der Übersetzung zu Komposita zusammengefügt werden sollen, treffen wir erstmals unter Berücksichtigung von syntaktischen Informationen, die aus dem zu übersetzenden Satz aus der Quellsprache (in diesem Fall: Englisch) abgeleitet wurden.

Wir überprüfen die Qualität unseres morphologischen Ansatzes zur Kompositazerlegung einerseits anhand manueller Evaluierungen, andererseits messen wir den Einfluß unserer Kompositabehandlung auf die Übersetzungsqualität eines aktuellen, frei verfügbaren, SMÜ Systems. Wir untersuchen beide Übersetzungsrichtungen des Sprachpaares Deutsch und Englisch. Wo immer möglich, vergleichen wir unsere Ergebnisse mit früheren Ansätzen zur Kompositabehandlung, die zumeist ohne morphologisches Wissen auskommen.

Acknowledgements

My deepest thanks to Alex Fraser, for being the best *handledare* one could wish for on the journey which led to this thesis. Through hiring me in the Morphosyntax-project, he allowed me to combine much of the thesis work with the project work. I thank him for always having an open door and for spending so much time with me and my thesis. I am most grateful to Alex for his optimism in judging intermediate results, for teaching me *ACLish* thinking and for never losing faith in me.

I am grateful to Jonas Kuhn for being a great *Adoptivdoktorvater*. He almost naturally took over as a main examiner when the bureaucratique circumstances required me to find a new main one. I thank him for his interest in my work, many constructive discussions and for integrating me into his group.

I thank Jörg Tiedemann for having agreed to review my work and for pointing me to *Ungereimtheiten* I would otherwise probably never have discovered.

Thanks to all "old" and "new" professors and secretaries of IMS for making this such a great place to study and graduate. Thereof especially Hinrich Schütze for ensuring me funding and for helping me reconcile my work with the rest of my life. I am grateful to Ulrich Heid for having *angesteck*$_{<DE><V>}$*ed*$_{<EN><Suff>}$ me with his enthusiasm for morphology since the very first lecture in my undergraduate studies and for happily discussing morphological cases of doubt with me ever since. I would like to thank Helmut Schmid for instant SMOR support whenever required and Edgar Hoch and his system administration crew for saving me from countless computer-related emergency situations. Special thanks to Sabine Dieterle – the good soul of IMS – for leading me through the jungle of German university bureaucracy.

Dating back to my time as an undergraduate *Hiwi* at IMS, I started working with Marion Weller, and have been doing so ever since. It has been a great pleasure and a privilege to have worked with Marion for so long and I hope that our common *Freizeitforschungsinteresse* will occasionally bring us together again in the future.

During my PhD, I learned that nothing improves performance as much as a good lunch experience. I am very thankful to the phonetics group at IMS for integrating me into their *Mittagessensrunde* and thereby distracting me from BLEU scores for an hour each day to discuss topics that really matter. Thereof thanks Nadja for selecting unforgetable topics, Kati for constantly extending my knowledge in the art of consuming raw vegetables, Antje for giving me that one most important ride home and Mike for occassionally putting an *"orange"* on the table.

As for other colleagues at IMS, I thank Christian Scheible for being more than just my *Charles* and André Blessing for mental and Linux support. Moreover, I am grateful to Boris Haselbach for always being available to discuss urgent (semi-)important questions on the hallway and to my office mate Anita Ramm for listening, discussing, and distracting, doing so with and without chocolate – and for wisely choosing among these options. Other companions at IMS include Kerstin Eckart, Nina Seemann, Fabienne Braune, Daniel Quernheim, Anders Björkelund, Wiltrud Kessler and Stefanie Anstein. Outside IMS I would like to thank Sara Stymne, in whose footsteps I started doing my research, Baki Cakici for down-to-earth discussions, Katharina for sharing the burden of everyday life with me and Karoline, Tim, Anouk and Inga for their patience regarding my absences and unavailabilities. Finally, I thank the participants of Dagstuhl Seminar Nr. 14061 for a week full of inspirations. It has been a long long time ago, but I can still remember!

I thank Sabine Schulte im Walde, Heike Zinsmeister and Magnus Boman for being role models I have been looking up to, and Alba, Johanna and Müjde for being the greatest in their respective jobs.

Herzlichste thanks go to Magnus and Eero for constantly reminding me of the fact that **this is *just* a thesis (!)**. There is so much more to life and time can be wasted so much more gladly! Last and most, I thank Oliver for all of his love, patience and support and for always flying at my side – all the way through.

The work in this thesis was supported by the DFG grant Models of Morphosyntax for Statistical Machine Translation.

Contents

List of Abbreviations

BLEU	BiLingual Evaluation Understudy, (Papineni et al., 2002)
CRF	Conditional Random Field
FST	Finite-State Transducer
MERT	Minimum Error Rate Training
METEOR	Metric for Evaluation of Translation with Explicit ORdering (Lavie and Agarwal, 2007)
MT	Machine Translation
NLP	Natural Language Processing
PBSMT	Phrase-Based Statistical Machine Translation
POS	Part of Speech
SMOR	Stuttgart MORphological Analyser (Schmid et al., 2004)
SMT	Statistical Machine Translation
SVM	Support Vector Machine

1. Introduction

Machine Translation denotes the translation of a text written in one language into another language performed by a computer program. It enables access to texts written in a foreign language, without knowing anything about that language. Having emerged from military interests, the application range for machine translation has meanwhile – in our times of globalisation and the growing importance of the internet – expanded to civil applications of everyday life. Many of today's state-of-the-art machine translation systems are based on statistical models. These are trained on a large text collection in the two languages of the translation pair. In this thesis, we investigate whether the translation quality of such a statistical machine translation (SMT) system can be improved using compound processing. In contrast to most previous works, we use linguistic knowledge to preprocess the underlying text collection on which the SMT system is trained.

1.1. Motivation

The underlying idea of **Statistical Machine Translation** (SMT) is to learn translational equivalences based on a large bilingual text collection. This text must be parallel, i.e. each line in one language must correspond to the translation of that line in the other language. This **parallel training data** is usually taken from existing human translations in order to ensure a high translation quality. Different statistical models are then trained on the parallel training data, e.g. based on co-occurrence frequencies of words occurring in the same lines of both sections of the parallel data. These models allow the SMT system to translate any sentences, as long as the words they consist of have occured in the parallel training data. **Words that have not occurred in the data cannot be translated**. Instead, they are transferred as they are, in their original language.

German is a **morphologically rich language**: depending on their context, words may occur in different inflectional variants. Moreover, German also allows for **productive compounding**, i.e. the creation of new complex words based on a concatenation of simple words. As a consequence of this variety and productivity, SMT systems designed for translation from and to a morphologically rich language like German often suffer from **data sparsity**: some of the words to be translated might not have occurred (or at least not sufficiently often) in the parallel training data and can thus not be translated.

Nevertheless, the training data often provides enough information to translate unseen compounds: while many compounds might not have occurred, their component words usually have occurred. **Separating compounds into their component words** prior to the translation process, makes them translatable part-by-part. If, in addition to that, the training data is **lemmatised**, the translation model can **abstract over different inflectional variants**. For the English to German translation direction, the compound processing requires a post-processing step in which simple words are merged into compounds and inflectional endings are predicted.

While compound splitting has become state-of-the-art in German to English SMT, the opposite translation direction, from English to German has received much less attention in the past. To our knowledge, there is currently **no other system that combines compound processing with inflection handling** for English to German SMT.

1.2. Contributions

The subject of this thesis is to integrate compound processing into SMT for the language pair of English and German. As parallel training data is limited, SMT systems must make the most out of the information encoded in the available data.

I present a **compound processing system** for statistical machine translation (from and to German and English) that incorporates **linguistic knowledge** from a rule-based morphological analyser. I will combine compound processing with inflection handling in order to allow for a maximal generalisation over the training data.

German to English For this translation direction, my compound processing provides **highly accurate splittings** into component words. This enables part-by-part translations of compound words and reduced the number of unknown words. As a consequence, **translation quality improves significantly**, even with respect to previous, linguistically less informed, approaches.

English to German For this translation direction, I combined my compound processing system with an already existing inflection component, which enables not only the creation of **new compounds**, but also unseen **inflectional variants** thereof. The usage of a rule-based morphology allows for a **free combination** of former compound parts and simple words and allows for maximal generalisation. Moreover, translation quality improves and I can show that more compounds are produced correctly. In contrast to previous work, I use **syntactic information** derived from the English **source language** to decide on compound merging.

Methodology I compare the impact of using a morphological analyser for compound processing with re-implementations of widely used previous approaches. While my main focus is on improving end-to-end statistical machine translation, I also report on clean data experiments and detailed error analyses I performed.

1.2.1. Secondary Contributions

Awareness The problem of productive compounding has been investigated by numerous research groups for translations from German into English before. In the opposite translation direction (from English into German), however, it has not yet received much attention from the SMT community. My thesis explicitly addresses productive compounding for translating **into** German and thus **raises awareness** for this problem in the SMT community.

Gold Standards In order to evaluate the accuracy of my compound processing approaches and compare it to the performance of previous approaches, we created numerous **gold standards**. These will be made **publicly available** in order to make my results comparable to future approaches.

Extrinsic Evaluation The intrinsic evaluation of rule-based morphological analysers in terms of coverage is a non-trivial task and requires large amounts of manually annotated data. However, using a rule-based morphological analyser to improve end-to-end statistical machine translation can be considered a successful **extrinsic evaluation of the morphological analyser**. The extrinsic evaluation of language resources is an ongoing challenge in the communities working on the creation of such resources.

1.3. Road Map

This thesis is divided into five parts (Background, Compound Splitting, Compound Merging, The Bottom Line and the Appendix). In the following, we briefly describe the content of each part.

I. **Background**, where we define German compounds and show examples (Chapter 2), present details concerning morphological analysers (Chapter 3) and briefly review statistical machine translation (Chapter 4). We further motivate the usefulness of compound processing in statistical machine translation and show how it can be integrated into a standard SMT system (Chapter 5).

II. **Compound Splitting**, where we describe two commonly used previous approaches to compound splitting (Chapter 6), before we present our morphology-based compound splitting in detail (Chapter 7). Then, we evaluate these three approaches with respect to manual compound gold standards (Chapter 8). Finally we integrate them into an end-to-end German to English SMT system in order to compare their impact on translation quality (Chapter 9). This part closes with a review of related works (Chapter 10).

III. **Compound Merging**, where we present our approach to compound merging and combine it with an already existing inflection prediction component (Chapter 11). We evaluate the accurracies of different feature combinations for compound merging on clean data (Chapter 12). We integrate the whole compound merging procedure into an end-to-end English to German SMT system (Chapter 13). Finally, we give a retrospective of related works (Chapter 14).

IV. **The Bottom Line**, where we conclude our findings and summarise our contributions. We discuss the shortcomings of our work and give some directions for future work (Chapter 15).

V. **Appendix**, where we informally compare the rule-based morphological analyser we use with another existing analyser. Moreover, the appendix contains additional results of the gold standard evaluations and the SMT experiments.

More detailed information on the contents of parts I.-III. and their chapters are given at the beginning of the respective parts.

Part I.

Background

Motivation In the first part of this thesis, we provide some background knowledge for the contents that will be addressed in the subsequent parts. Recall that the focus of this thesis is on compound processing for Statistical Machine Translation (SMT). The language pair under investigation is English and German, we thus begin with an introduction to productive German compounding. For the processing of compounds, we will make use of a rule-based morphological analyser. We describe the general concepts of such analysers, together with details on the analyser we will use. Moreover, we introduce the basic components of a classical statistical machine translation system, which will remain unchanged throughout our experiments. Finally, we motivate the intuitive idea of compound processing for SMT for translation from and to English and German.

Contributions We describe how typical characteristics of German compounds lead to data sparsity issues in SMT and motivate how some of these issues can be solved by using compound processing. In the past, most of the compound processing approaches for SMT were **not** based on rule-based morphological analysers. Moreover, most of the available approaches were designed for translation from German into English. In the opposite translation direction, from English into German, the usage of a rule-based morphological analyser for compound processing has some clear advantages. For example, compound splitting and lemmatisation can happen in one joint step. Due to morphological analysers working bidirectionally, compounds can be merged and inflected in one joint step, too. This thesis raises awareness for the usefulness of morphological analysers to perform compound processing for SMT.

Structure The remainder of this part is structured as follows: In Chapter 2, we introduce characteristics of German productive compounding. Then, we describe morphological analysers in Chapter 3. First, we focus on their general architecture and then we give details on the analyser we are using. In Chapter 4 we describe the origins of machine translation in general, and go into details of Statistical Machine Translation. Finally, in Chapter 5, we bring it all together and motivate the benefits of using compound processing in SMT.

2. About German Compounds

The focus of this chapter is on German compounds. Due to their productivity, German compounds are challenging for data-driven applications for Natural Language Processing (Nlp), including statistical machine translation. In this chapter, we will take a closer look at the characteristics of German compounds and give a short outlook on how these will be handled in our compound processing system.

Terminology A German compound usually consists of two (or more) simple words that have been put together to form a new word. The rightmost part of the compound is referred to as the compound *head*, whereas all other parts are called *modifiers*. Almost all German compounds are right-headed, i.e. with the rightmost part denoting the head. Theoretically, a German compound could have an unlimited number of modifiers, whereas the vast majority of compounds has only one head.

Structure The main characteristics of German compounds and how these are handled by our compound processing approach are discussed in Section 2.1. Section 2.2 deals with filler letters (called *"Fugenelemente"*) that are often required to build German compounds. The portemanteaus introduced in Section 2.3 can be considered a special case of German compounds. Compounding languages other than German are considered in Section 2.4. A summary of this chapter is given in Section 2.5.

2.1. Features of German Compounds

In this section, we describe some characteristics of German compounds, ranging from *productivity* and *complexity* over *frequency* to *compositionality* and *lexicalisation*, and then discuss how they will be handled by our compound processing system.

Productivity Compounding is a highly productive and creative word formation process in German: new compounds can be generated from scratch and there are hardly any limitations which words to combine.

head → **↓ modifier**	**noun**	**verb**	**adjective**
noun	Hausboot *house boat* Haus + Boot *house+boat*	Kopfrechnen *mental arithmetic* Kopf+rechnen *head+to calculate*	Kegelförmig *cone shaped* Kegel+förmig *cone+shaped*
verb	Schlaftablette *sleeping pill* schlafen+Tablette *to sleep+pill*	gefriertrocknen *to freeze-dry* gefrieren+trocknen *to freeze+to dry*	abholbereit *ready for collection* abholen+bereit *to collect+ready*
adjective	Blaulicht *blue light* blau+Licht *blue+light*	tieftauchen *deep diving* tief+tauchen *deep+dive*	frühkindlich *early infantile* früh+kindlich *early+infantile*

Table 2.1.: Examples of different POS-patterns for German compounding.

Most German compounds consist of nouns, but even adjectives, adverbs and verbs can be productively combined to form new compounds. In Table 2.1 we give examples of frequently occurring POS-patterns (= part-of-speech) of German compounds. Apart from these, even adverbs (*"Außenantenne"* = "exterior antenna"), numerals (*"Viergespann"* = "carriage and four"), prepositions (*"Nebenzimmer"* = "adjoining room") and pronouns (*"Niemandsland"* = "no man's land") are occasionally found in compounds (examples taken from Fleischer and Barz (1995), p. 113–120). However, these are much less productive than nouns, adjectives and verbs.

Complexity While the compounds given in Table 2.1 consist of two parts, there is theoretically no limit to the number of words that can be combined into a noun compound. The internal structure of such n-ary compounds (with n>2) is determined by the semantics of the compound. For example, *"Schmerz|mittel|klasse"* (lit. = "pain|middle|class") denotes a class of pain killers [[Schmerz|Mittel]Klasse] instead of being a painful middle class: [Schmerz[Mittel|Klasse]]. In German left-branching structures are more common than right-branching structures, and there are also some indecisive cases, where the branching might be determined based on context. Illustrative examples for different branching structures are given in Figure 2.1. While *"Hausstauballergie"* (= "house dust allergy") is an allergy against house dust, the right-branching *"Seidenhaarband"* (= "silk hair ribbon") is not a ribbon for silk hair, but instead a hair ribbon made of silk. The

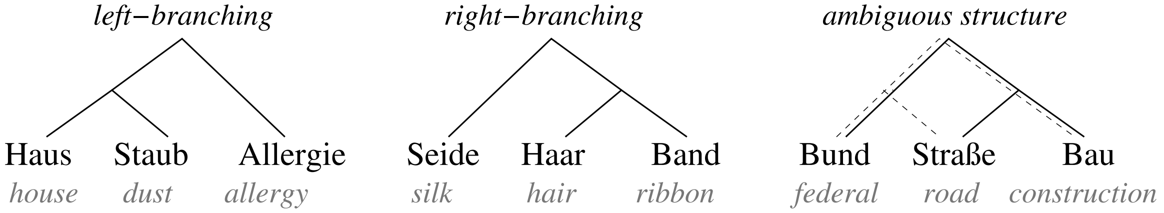


Figure 2.1.: Illustration of left- and right-branching structures of German compounds, with the rightmost example being structurally ambiguous.

case of *"Bundesstraßenbau"* (= "federal road construction") is ambiguous: it can either denote the construction of federal roads (highways) or the federal construction of roads.

With more than three compound parts, the internal structure gets even more complex. In Figure 2.2, we give a real-world (!) example of a German law (repealed in 2013): *"Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz"* (= "beef|labelling|supervision|duty|delegation|law"), a law on the delegation of duties for the supervision of beef labelling.[1]

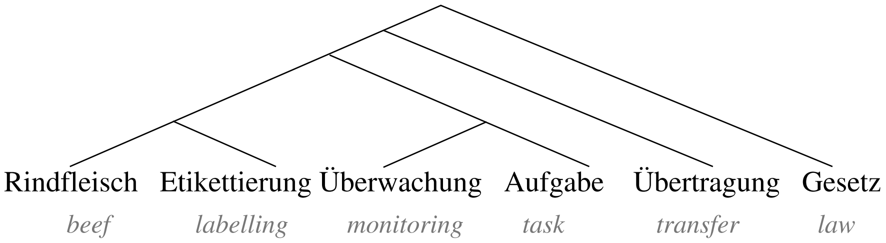


Figure 2.2.: Internal structure of the compound *"Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz"*, a law on the delegation of duties for supervision of beef labelling.

Frequency Compounds occur in any kind of German text, be it newspaper or text from a specific domain. Generally speaking there exist a huge number of different compounds, but only a small subset of them occurs reccurrently. According to Baroni et al. (2002), compounds make up only 7% of the token words of a 28 million word corpus but 47% of the word types. Most of the compounds they found (83%) occured only 5 times or less. Monz and de Rijke (2001) report on the proportions of compounds in a much smaller German corpus consisting of roughly 3,000 words. They found that 91% of all compounds

[1] In German: *"Dieses Gesetz regelt die Übertragung der Überwachungsaufgaben der Etikettierung von Rindfleisch."*

group	**description**	**example** *gloss*
endocentric	The modifier specifies the head word *Messer*.	Brotmesser *bread knife*
subcategorised	The modifier fills the subcategorisation frame of the head word, which is often derived from a transitive verb (here: to drive).	Busfahrer *bus driver*
copulative	The modifier and head are a coordination on the same hierarchical level, the order can be reversed without changing the semantics of the compound.	nasskalt *chilly and damp*
exocentric	The semantic head of the compound is not part of the compound: our example denotes a **motor** with four cylinders.	Vierzylinder *four cylinders*

Table 2.2.: Classification of compositional German compounds.

consist of two words, 8.5% of three words, and 0.5% of four parts or more. In domain specific texts, the proportions of compounds is higher in general. Usually, there are also more n-ary compounds. Based on a 20 million token text from the computer domain, Marek (2006) found that 40% of all word types were compounds: thereof, 83% consisted of two parts, 16% of three parts and 1 % of four and more parts.

The diversity of compounds on the one hand, which results in a large number of rare compounds and their overall presence on the other hand makes them a serious problem to be dealt with by most NLP applications.

Compositionality So far, we described the productivity, complexity and frequency of compounds. In this paragraph, we address their semantics: there are *compositional* compounds, whose meaning can be derived from the meaning of the component words and *non-compositional* compounds, where the meaning is less (or not anymore) related to the meaning of their component words.

Compositional German compounds can be further classified into four groups, depending on the relation between the modifier and the head word: i) endocentric compounds, ii) subcategorised compounds, iii) copulative compounds and iv) exocentric compounds.[2] A short description of each of these groups is given in Table 2.2, together with an example.

However, the meaning of a compound is not always determined by the meaning of

[2] Correspondence to German terminology: *Determinativkomposita* = endocentric compounds, *Rektionskomposita* = subcategorised compounds, *Kopulativkomposita* = copulative compounds and *Possessivkomposita* = exocentric compounds.

its parts. Consider e.g. *"Heckenschütze"* (= "sniper", lit. "hedge shooter"): most often, a sniper is a kind of shooter, but he need not neccessarily shoot from behind a hedge. In the German compound, the hedge is an indicator for a hidden attack, which is typical for a sniper. While the semantics of *"Heckenschütze"* is semi-compositional (at least, it is a kind of shooter), there are also more opaque (= non-compositional) compounds like *"Kotflügel"* (= "mudguard", lit. "droppings wing"), which denotes neither droppings nor a wing. Taking a look at the etymology of this word, one finds that it has been used to protect carriages from mud[3] and that early versions resembled wings of a bird.

In the course of creating the gold standards (cf. Chapter 8), we found that the vast majority of German compounds are compositional. It seems as if compositional compounds are much more productive than non-compositional compounds. However, non-compositional compounds often had a compositional origin (e.g. *"Heckenschütze"*, *"Kotflügel"*) which gets lost over time.

Context-Dependence Sometimes, the analysis of a compound is dependent on the context in which it has occurred. Consider for example the compound *"Samthandschuhe"* (= "velvet gloves"). It mostly occurs in idiomatic expressions like *"jmd. mit Samthandschuhen anfassen"* (= "to handle s.o. with kid gloves", lit. "to touch s.o. with velvet gloves") or *"die Samthandschuhe ausziehen"* (= "start to playing rudely", "to take off the velvet gloves"). In these contexts, the meaning of the compound, being integrated into an idiomatic expression, is non-compositional. However, archivists working with very ancient books or paper fragments often litteraly wear velvet gloves. In such contexts, the compound is to be interpreted compositionally. In addition to that, some German compounds coincide with German proper names. For example, *"Baumeister"* may be interpreted as "master builder" or as a proper name. In the latter case, it should not be split.

Moreover, even the way a compound is split is sometimes dependent on the context in which it has occurred. For example, the German compound *"Wachstube"* can either be split into *"Wach|Stube"*, where it means "guard|room", or into *"Wachs|Tube"*, where it means "wax|tube". The appropriate splitting is determined by the context of the word. Such compounds are called parasite words (Airio (2006)).

[3] *"Kot"* is used for "droppings" in modern German, but its original meaning is "mud".

Lexicalisation Irrespective of their compositionality, some compounds are lexicalised: they are no longer perceived as consisting of two (or more) simple words, but as a simple word themselves. Examples include *“Lastwagen”* (= “lorry”, lit.: “load waggon”) and *“Bahnhof”* (= “station”, lit.: “train yard”). Most non-compositional compounds are lexicalised, but even compositional compounds that are very frequently used become lexicalised over time.

Relation to Our Compound Processing Approach The compound processing we perform addressed all of the above mentioned characteristics of compounds in the following way:

productivity	Newly created compounds can be processed by our system, even though they might never have occurred in a text before, as long as their component words are in the lexicon of our rule-based morphology.
complexity	The compound analysis of the rule-based morphology produces flat representations, no distinction is made between left- and right-branching strucutures.
frequency	If a compound occurs more frequently than the geometric mean score of the frequency of its parts, it remains unsplit.
compositionality	This is not yet considered in the current compound splitting, but for compound merging we use a fall-back feature to merge compounds that should not have been split in the first place.
context-dependence	In our splitting approach, we perform a token-based splitting. We take into consideration the POS of a word in order to prevent proper names from being split.
lexicalisation	In some of our experiments, we do not split compounds that are lexicalised in the rule-based morphology, even if their component words are still identifiable.

2.2. Fugenelemente

Fugenelemente (short: *Fugen*), also named *filler letters*, *linking morphemes* or *compounding suffixes* may sometimes need to be introduced between two simple words in order to form a German compound. Aside from a few phonological requirements, there are no general rules for the introduction of these elements. In the past, two controversial opinions have emerged about the status of *Fugen* in compounding: whether they are i) a morpheme on their own, acting as an interfix between two simple words, or ii) whether they belong to the modifier of the compound or whether words might have a compounding stem form (including the *Fuge*). In the following, we consider both of these approaches and close the section with a quantitative overview of the most frequent German *Fugen*.

Independent Morpheme In Bergenholtz and Mugdan (1979), *Fugen* are considered morphemes with different possible forms (= *allomorphs*). They observe that the combination of a modifier with a *Fuge* sometimes coincides with inflection of the modifier, e.g. its genitive or plural form. However, due to the fact that these forms do not carry a genitive or plural interpretation when they occur in a compound, (e.g. *"Hühnerei"* (= "chicken$_{Pl}$ egg") is the egg of only one chicken), *Fugen* are considered to be independent morphemes instead of modifiers occurring in their inflected form.

Part of the Modifier Both Fuhrhop (1996) and Langer (1998) consider *Fugen* to belong to the modifier of the compound (Langer (1998) calls them "compounding suffixes"). Fuhrhop (1996) distinguishes between **paradigmic** *Fugen*, which coincide with an inflectional ending of the same word, and **non-paradigmic** *Fugen*, where this is not the case. An example for the latter group is *"Geschichtsbuch"* = "history book", where *"Geschichts"* does not coincide with any inflectional form of the noun *"Geschichte"*.

Compounding Stem Form Due to the difficulties of defining general rules for the insertion of *Fugen*, Fuhrhop (1998) introduces the notion of compounding stem forms (consisting of the stem and the *Fuge*), which are part of a stem paradigm. They must be defined separately for each stem, and there might be more than one compounding stem form for the same stem. Heid et al. (2002) adopt this concept and terminology for the development of DeKo, which is one of the antecedent systems on which SMOR (Schmid et al., 2004), the rule-based morphology we use, is based.

2.3. Portemanteaus

A portemanteau is a single morphological unit, which has been built from (at least) two morphemes. Sometimes, the original morphemes cannot longer be distinguished, e.g. French *"à+le = au"* ("to+the"). German allows for the productive formation of portemanteaus from a verb and a pronoun in the spoken language, e.g. *"ist + er = isser"*. In written German however, portemanteaus are restricted to a combination of prepositions and definite articles, e.g. *"in+dem = im"* (= "in+the"). Here, the case of the definite article must agree with the noun to which it is directed. As portemanteaus introduce additional sparsity with respect to machine translation, they will be addressed in our English to German system.

2.4. Other Languages

Compounds do not only occur in German, but also in a number of other languages. However, compounds are often written as separate words with whitespaces in between as in English: "sewing machine"[4] or in slavic languages like Russian *"švjnaja mašina"* or Croatian "šivaća mašina". In Romance languages, the component words are often connected with prepositions like in French: *"machine à coudre"* or Spanish *"máquina de coser"*. Compounds without whitespaces in between (sometimes called "closed compounds") are a phenomenon that mostly occurs in Germanic languages like Swedish and Norwegian *"sy+maskin = symaskin"*, Danish *"sy + maskine = maskine"*, Dutch *"naaien+machine = naaimachine"*. Beyond Germanic languages, also Finnish *"ompelu+kone = ompelukone"* and Hungarian *"varrni + gép = varrógép"* allow for closed compounds.

In Arabic, closed compounds are limited to a combination of the stem with certain affixes which are used to express, for example, prounous, prepositions and conjunctions. The Arabic language also features productive compounds with whitespaces in between. For these, a genitive construction is used, similar to compounding in romance languages.

Finally, even in languages that do not allow for the productive formation of closed compounds, one can find some exceptions. Examples include the English *"flowerpot"*, *"gentleman"* and the French *"portefeuille"* (= "wallet", lit: "carry leaves") and *"bonhomme"* (= "man", lit. "gentle man").

[4] We use the example of "sewing machine" to illustrate differences in compounding across languages. Some of these examples are taken from Bergenholtz and Mugdan (1979), p.175.

2.5. Chapter Summary

In this chapter, we shed some light on the morphological phenomenon of compounds in German. We discussed their characteristics, amongst others in terms of productivity, their complexity, and their semantic interpretation. Along the way, we gave numerous illustrative examples. Finally, this chapter presented some of the theoretical background on which the next chapter about morphological analysers will be built.

3. Morphological Analysers

One of the research questions that this thesis aims to answer is whether linguistic knowledge derived from a rule-based morphological analyser improves compound processing for SMT. Before we go into the details of how to perform compound processing (cf. Chapter 7), we will introduce morphological analysers in general and describe the one we use throughout our work, namely Smor (Schmid et al., 2004).

Structure The chapter is structured as follows: we introduce some theoretical and technical background in Section 3.1. Based on that, we present a detailed description of Smor in Section 3.2, including examples on how it performs compounding, derivation and inflection. In Section 3.3, we take a look at Gertwol, another rule-based morphological analyser which has been widely used in NLP applications before. In Section 3.4, we mention advantages and disadvantages of morphological analysers before we conclude the chapter with a summary in Section 3.5.

3.1. Background

This section introduces some theoretical and technical background on which the rule-based morphology Smor is built. It consists of two parts: we first present Two-level Morphology (Section 3.1.1), which is a simple formalism for morphological descriptions and then we give a very basic introduction to finite-state transducers in Section 3.1.2.

3.1.1. Two-level Morphology

In the 1980s, Koskenniemi (1983) introduced a formalism to describe complex morphological phenomena, called *two-level morphology.* It was the first model which allowed to efficiently implement morphological analysers using finite-state technology. The key concept is to simultaneously use two levels of morphological description during the analysis:

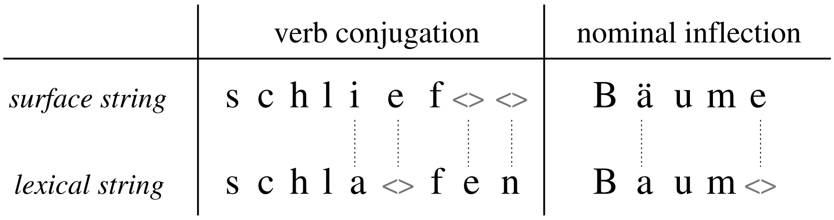


Figure 3.1.: Example for inflectional processes modeled with two-level-morphology.

i) the surface level, which is the full word form to be analysed and ii) the lexical level, which is the analysed (decomposed and lemmatised) word form.

The two-level formalism requires no intermediate representation to map the input to the output string. Thus, (parts of) the morphological analysis can be encoded directly in the lexicon of a finite-state based morphological analyser, e.g. the *Ablaut* often occurring in verb conjugation or morpho-phonological processes like the *Umlautung*. Examples are given in Figure 3.1. Constraints on morphological operations can be imposed on either (or both) levels of representation, as all modifications of the input string happen simultaneously. Previous approaches realised every modification of a word using rules, which made the creation of the analyser and also the processing much more time-consuming. See Karttunen and Beesley (2001) for more details on two-level morphology.

3.1.2. Finite-State Transducers

A finite state automaton is a conceptual model that describes the processing of input symbols using a finite number of different states and a set of possible transitions between these states. The formalism allows the definition of a finite-state automaton for any regular expression. This automaton then accepts all words that belong to the regular language the expression. Many NLP applications can be addressed using regular languages and efficiently be implemented using finite-state technology, e.g. morphology or parsing (Karttunen, 2001).

While finite-state **automata** process a sequence of input symbols which they either accept or not, a deterministic finite state **transducer** (FST) does not only process the input but also generates exactly one output symbol for each input symbol it reads. FSTs are thus a suitable mechanism to model morphological processes. In Figure 3.2, we give a graphical illustration of an FST for lemmatisation of the singular and plural form of *"Baum"* (= "tree"). As can be seen, this transducer has exactly one start and at least

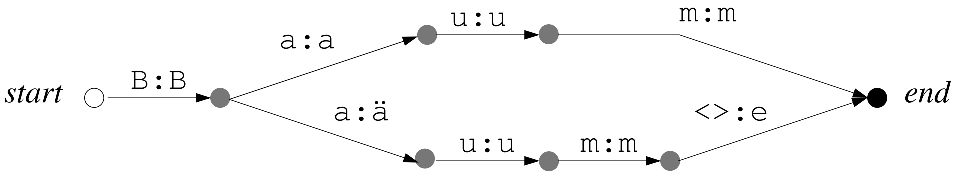


Figure 3.2.: Illustration of an simple finite-state transducer that analyses the singular form *"Baum"* (= "tree") and its plural form *"Bäume"* (= "trees") and maps them to their lemma, *"Baum"*. "<>" denotes an empty string.

one end state, which are formal requirements all finite-state transducers must fulfil. The arcs visualise the operations required – on the input and output string respectively – in order to move from one state to the next. The left and right hand side of the colons correspond to the two levels of morphological description: the surface string to be analysed is located on the right side and the analysed ("lexical") string is on the left side of the colon. Note that the morphological analysis of a finite-state transducer works in both directions: words can be analysed in one direction, but also generated, when the transducer is applied in the opposite direction. The two-level formalism for morphological description supports this property. According to (Koskenniemi and Haapalainen, 1996, p.133),

> *"Bidirectionality has always been one of the fundamental principles of the two-level morphology."*

In this thesis, we will make use of both directions: we use the analysis direction for compound splitting in Chapter 7 and the generation direction for compound and inflectional generation in Chapter 11.

3.2. SMOR

In the previous section, we introduced the theoretical background required to adequately describe Smor, including morphological models, the concept of two-level morphologies and some basics on finite-state technology.

Smor is a rule-based morphological analyser for German, covering inflection, compounding and derivation (Schmid et al., 2004). Following the morphological theory of *item and arragement* (Bergenholtz and Mugdan, 1979), word formation in Smor is considered to be the result of a concatenation of free morphemes with other free or bound

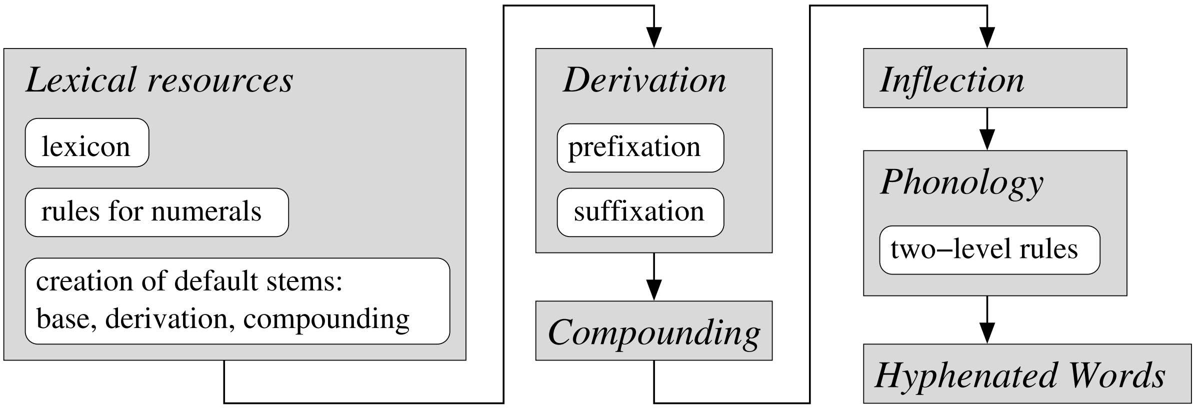


Figure 3.3.: Simplified schematic illustration of Smors components.

morphemes, for example derivational or inflectional morphemes. In terms of implementation, Smor is a *two-level morphology*, realised as a *finite-state transducer*. This allows to process the input and output string simultaneously and makes the analysis direction reversible. Moreover it enables a straightforward and efficient modeling of morphophonological processes like e.g. *"Umlaut"*, which is in line with the theory of *item and process*.

Finally, the analysis of compounds is implemented based on the concept of compounding stems as opposed to filler letters (with very few exceptions).

System Architecture As previously mentioned, Smor is implemented as a finite-state transducer (FST). The main transducer incorporates several smaller FSTs which handle certain sub-tasks of the analysis process. The division into multiple transducers simplifies maintenance: for example, it enhances the compilation speed of the main transducer if unmodified sub-transducers can be pre-compiled. An illustration of the subtasks of the morphological analysis in Smor (in the order of their use in an analysis process) is given in Figure 3.3. Note that not each of these subtasks is represented by a separate FST.

Availability As of July 2014, Smor is freely available for research purposes through the CLARIN-D center located at the Institute for Natural Language Processing in Stuttgart, Germany.[5] The lexicon can exclusively been obtained from CLARIN for research purposes, whereas the finite-state tools and compilers are open source and can be down-

[5] CLARIN is a European project currently building a **C**ommon **La**nguage **R**esources and Technology **In**frastructure, see `http://clarin-d.org/de` (as of July 2014) for more details.

category	**possible tags**
Entry type:	<Stem> <Suffix> <Prefix>
Stem type:	<base> <deriv> <compound>
Word class:	<V> <ADJ> <NN>...
Origin:	<native> <foreign> <classical>...
Complexity:	<simplex> <prefderiv> <suffderiv>
Inflectional Class:	<Adj+> <NFem-Deriv>

Table 3.1.: Examples for description categories of SMOR lexicon entries, (Taken from (Schmid et al., 2004, p.1263)).

loaded from the webpage of the developer, Helmut Schmid.[6]

Structure In the following, we discuss the importance of the lexicon in Section 3.2.1. In Section 3.2.2, we explain how compounding is modeled in SMOR. Then, we briefly present derivational processes in Section 3.2.3 and inflection in Section 3.2.4.

3.2.1. Lexicon

In order to perform word formation, derivation, and inflection properly, the lexicon of a rule-based morphological analyser must consist of more than just naked stem and affix entries. In SMOR, the properties of the lexical entries are encoded with a set of features. Some lexical features, taken from Schmid et al. (2004), are given in Table 3.1.

As can be seen, three different *entry types* are distinguished: stems, suffixes, and prefixes. For compounding, our main focus is on stems. SMOR's lexicon features three different *stem types*: base stems, derivation stems and compounding stems. Note, however, that the default word formation process in SMOR is rule-based. The compounding stems which are directly encoded in the lexicon are exceptions to these rules.

The *origin* and *complexity* features given in Table 3.1 are used to model derivational constraints on stems and affixes (see Section 3.2.3 for details). Finally, each base stem is assigned to a word class and an *inflectional class* which produces the inflectional endings for each case, number, gender, and person variation of the lemma (see Section 3.2.4 for an example). For rule-based word formation, the inflectional class plays a crucial role: For example, compound modifiers often coincide with the singular genitive or plural nominative form of a lemma. SMOR takes advantage of this coincidence from an engi-

[6] `http://www.cis.uni-muenchen.de/~schmid/tools/SFST/`, as of July 2014.

	stem type	**part-of-speech**						
stems: 49,942	**base**	NN	ADJ	V	ADV	NE	ABR	ETC
	47,667	18,629	8,941	8,776	1,214	8,541	980	586
	derivation	NN	ADJ	V		NE		
	1,738	825	150	750		13		
	compounding	NN	ADJ	V	ADV	NE		ETC
	537	201	26	141	3	46		120

Table 3.2.: SMORs lexicon consists of 50,914 entries, whereof 49,942 are stems, 49 are prefixes and 253 are suffixes. The table illustrates the quantitative distribution over different stem types and parts of speech. NN = nouns, ADJ = adjectives, V = verbs, ADV = adverbs, NE = named entities, ABR = abbreviations, ETC = other.

neering perspective. This is in line with Fuhrhop (1998), see also Section 2.2. No claims are made about a semantic interpretation of the compound modifier as genitive or plural form. A more detailed example will be given in the next section, which deals with compositionality.

Coverage The coverage of a rule-based morphology depends on its application domain. SMORs lexicon is a large linguistic resource that has been built and maintained at IMS for more than a decade. In Table 3.2, we give details on the lexicon size of the version we used for all experiments presented in this thesis (dating back to 2008). As can be seen, we counted the number of entries, different entry types, stem types and POS.

surface level		**lexical level**		**accept**
input string	**inflected form**	**lemma**	**features**	
Orts	Ort	Ort	<Nom><Sg>	no
	Ortes Orts		<Gen><Sg>	no **yes**
	Orte		<Nom><Sg>	no

Table 3.3.: Default compound stem modeling in SMOR, illustrated for *"Orts"* (= "location"), which is the modifier of the compound *"Ortszeit"* (= "local time"). The nominative and genitive singular and the nominative plural forms of the potential modifier are produced (using the inflection class assigned to the lexicon entry of the stem) and then matched against the input string.

entry	**lemma**	gloss	**pos**	**type**	**origin**	**inflection class**
<Stem>	Aktion	*action*	<NN>	<base>	<free>	<NFem_0_en>
<Stem>	Baum	*tree*	<NN>	<base>	<native>	<NMasc_es_$e>
<Stem>	Hilfe:s	*help*	<NN>	<compound>	<native>	
<Stem>	Hilfe	*help*	<NN>	<base>	<native>	<NFem_0_n>
<Stem>	Ka:ärte:<>	*card*	<NN>	<deriv>	<native>	
<Stem>	Karte	*card*	<NN>	<base>	<native>	<NFem_0_n>
<Stem>	Kirsche:<>	*cherry*	<NN>	<compound>	<native>	
<Stem>	Kirsche	*cherry*	<NN>	<base>	<native>	<NFem_0_n>
<Stem>	Organisation	*organisation*	<NN>	<base>	<free>	<NFem_0_en>
<Stem>	Ort	*location*	<NN>	<base>	<native>	<NMasc_es_e>
<Stem>	Plan	*plan*	<NN>	<base>	<native>	<NMasc_es_$e>
<Stem>	Zeit	*time*	<NN>	<base>	<native>	<NFem_0_en>

Table 3.4.: Example entries from Smors lexicon, with glosses. Note that only base stems (cf. *type* column) are inflected (thus assigned to inflection classes). Lemmas are written in two-level morphology format (analysis:surface level).

3.2.2. Compounding

Compounding plays a central role in this thesis, we will thus have a closer look at how compounding is modeled in Smor. In general, any two freely occuring base stems can be combined. In the lexicon, these are marked with the features *<base>* and *<free>*. Compounding in Smor differs only with respect to how the compound stems (= words in modifier position of a compound) are created, i.e. whether or not a filler letter is introduced. Based on the lexicon entries given in Table 3.4, we show three concrete examples of compound stem modeling in Smor, based on: i) default compounding, ii) exceptions to default compounding, and iii) exceptions explicitly encoded in the lexicon.

Default Compounding As mentioned in Section 2.2, compound modifiers often coincide with the nominative or genitive singular or the nominative plural form of a word. The modifiers can either be identical to the lemma or marked with an inflectional ending. Fuhrhop (1996) (p.528) calls such inflectional endings paradigmic filler letters. Other filler letters that do not coincide with freely occuring German words are called non-paradigmic filler letters. In Smor, the default rule to model compound stems (= find the modifier of a compound) is to check whether one of these inflected forms matches the word to be analysed. Consider for example the compound *"Ortszeit"* (= "local time"), and the lexicon entries of its two component words *"Ort"* (= "location") and *"Zeit"* (= "time") in Table 3.4. The nominative and genitive singular and the nominative plural form can be straightforwardly built from the lexicon entry using the inflection class of *"Ort"*, namely *<NMasc_ es_ e>*[7] (see Section 3.2.4 below for details on how inflectional endings are created). Then, SMOR checks for each of the inflected word forms whether it exactly matches with the current input string (here: *"Orts"*). This matching process is illustrated in Table 3.3. For the present example, it can be seen that one of the genitive singular forms (here: *"Orts"*) matches the input. However, the features assigned to the modifier in this intermediate analysis step will be filtered out later on in the analysis process. Only the part-of-speech remains attached to the modifier. For the sake of completeness, this is SMOR's analysis output for *"Ortszeit"*:

```
analyze> Ortszeit
Ort<NN>Zeit<+NN><Fem><Acc><Sg>
Ort<NN>Zeit<+NN><Fem><Gen><Sg>
Ort<NN>Zeit<+NN><Fem><Nom><Sg>
Ort<NN>Zeit<+NN><Fem><Dat><Sg>
```

Exceptions to Default Compounding The formation of German compounds does not always follow the scheme described in the previous paragraph. In some cases, a filler letter needs to be introduced to correctly model a well-formed compounding stem. In contrast to the *"s"* appearing in the above example *"Ortszeit"* (= "local time") which coincides with the genitive singular form of the word *"Ort"*, the *"s"* in *"Aktionsplan"* (= "plan of action") is attached to *"Aktion"* only for compounding. There is no inflected form of *"Aktion"* with this letter attached.[8] In German, there are numerous such cases. They can most easily be grouped together according to their ending, which is often a suffix like e.g. *"-ion"*. This is the complete list of word endings in SMOR for which an additional filler *"s"* is allowed to be introduced. The regular expression notation is used to efficiently encode different word endings in SMOR:

word endings = [hk]eit | [Aa]rbeit | ung | ion | mut | [dmn]acht | [sz]ucht | fahrt | flucht | [Gg]eburt | kunft | pflicht | schaft | schrift | sicht | schicht | falt | tät | [Gg]eduld

[7] The name of the nominal inflection class often reveals the genitive and plural form of a word. In the present case, we have a *masculine* noun, whose *genitive singular* is built by adding the inflectional ending "+es" and the *plural* is built by adding "+e".

[8] All singular forms of *"Aktion"* remain morphologically unmarked like the lemma, and the nominative plural is *"Aktionen"*

lexicon entry	**lexical level**	**surface level**
Hilf**e:s**	Hilfe<base>	Hilfs<compound>
Kirsch**e:<>**	Kirsche<base>	Kirsc**h**<compound>

Table 3.5.: Visualisation of the lexical and surface realisations of the compounding stems for *“Hilfe”* (= “help”) and *“Kirsche”* (= “cherry”).

Exceptions Encoded in the Lexicon Besides the two cases already mentioned, German word formation also allows for exceptions that do not occur sufficiently often to make the writing of rules describing them feasible. An example is *“Hilfe”* which has two possible compounding stems. The first one coincides with the nominative singular and is covered by the default compounding stem rule, e.g. *“Hilferuf”* (= “call for help”). The other one requires the substitution of the last letter *“e”* by *“s”*, e.g. *“Hilfsorganisation”* (= “aid organisation”). Another example is *“Kirsche”*. Here, the last letter of the lemma is deleted for compound formation, e.g. *“Kirsche”* (= “cherry”) + *“Baum”* (= “tree”) → *“Kirsc**hb**aum”* (= “cherry tree”). Such exceptions can directly be encoded in the lexicon, using two-level rules. Re-consider the compounding stem entries for *“Hilfe”* (= “help”) and *“Kirsche”* (= “cherry”) in Table 3.4 above. In Table 3.5, we give the lexical and surface realisations of these entries.

Issues We have already mentioned the ability of FST-based morphological analysers to process words in both, the **analysis** and the **generation** direction in Section 3.1.2 above. Like other morphological analysers, SMOR was mainly concieved to achieve high performance and coverage for the analysis of words, as this meets the requirements of most NLP applications. Regarding the generation of compounding stems, the rules and exceptions we introduced in the previous paragraphs sometimes lead to multiple surface realisations. Re-consider the previous examples, *“Hilferuf”* (= “call for help”), *“Hilfsorganisation”* (= “aid organisation”) and *“Kirschbaum”* (= “cherry tree”). In some of such cases, it might be possible to add some hand-written rules in order to prefer certain realisations depending on their context. However, adding more and more rules to the analyser not only makes its maintenance more difficult, but also slows down the analysis process. As with other NLP applications, a certain degree of ambiguity remains in a rule-based morphological analyser. See Table 3.6 for a complete list of over-generations for the compounding stems
of *“Hilfe”* (= “help”) and *“Kirsche”* (= “cherry”). Note that *“Kirschenbaum”* is not the

compounding stem (modifier)	**compound formation rule**	**decision**
Hilf**e**organisation *aid organisation*	default rule, nominative/genitive singular	wrong
Hilf**en**organisation *aid organisation*	default rule, nominative plural	wrong
Hilf**s**organisation *aid organisation*	exception encoded in the lexicon	**correct**
Hilf**e**ruf *call for help*	default rule, nominative/genitive singular	**correct**
Hilfen**en**ruf *call for help*	default rule, nominative plural	wrong
Hilf**s**ruf *call for help*	exception encoded in the lexicon	wrong
Kirsch**e**baum *cherry tree*	default rule, nominative/genitive singular	wrong
Kirsch**en**baum *cherry tree*	default rule, nominative plural	accept
Kirsc**hb**aum *cherry tree*	exception encoded in the lexicon	**correct**

Table 3.6.: Over-generation of compounding stems in Smor. The *decision* in the rightmost column is made from a linguistic point of view. Smor accepts all of these variants.

preferred, but an acceptable variant. In Section 11.5.2 we will explain how we select among multiple generated options in the course of our compound generation process.

3.2.3. Derivation

In this section, we briefly summarise derivation in Smor. Recall from Table 3.1 that two feature classes of the lexicon entries are used to model constraints for derivation: *Origin* and *Complexity*. Moreover, the POS of the stems with which an affix can be combined is also restricted. For example, only nouns that have a derivational stem entry in the lexicon can be combined with affixes. To illustrate how these constraints are used in practise, take a look at the following lexicon entries:[9]

1 <Stem><ge>**heile:**<>**n:**<> <V><base><native> <VVReg>
2 <Stem>**Ka:ärte:**<> <NN><deriv><native>
A <Suffix><simplex> <native><base><V> **bar**<ADJ><SUFF><base><native><Adj+>
B <Suffix><simplex> <native><deriv><NN> **chen**<NN><SUFF><base><native><NNeut_s_x>

They consist of two stem entries (1+2) and two suffix entries (A+B). We already introduced the feature format for the stem entries in Tables 3.1 and 3.4 above. The format of the suffix entries consists of two parts: the features to the left of the suffix string determine the kinds of stems the suffix can be attached to, whereas the features to the right of the suffix string describe the result of the affixation: e.g. the suffix *"-bar"* can be combined with a native base verb stem and if so, the result of this derivation is a

[9] This representation has been simplified for readability.

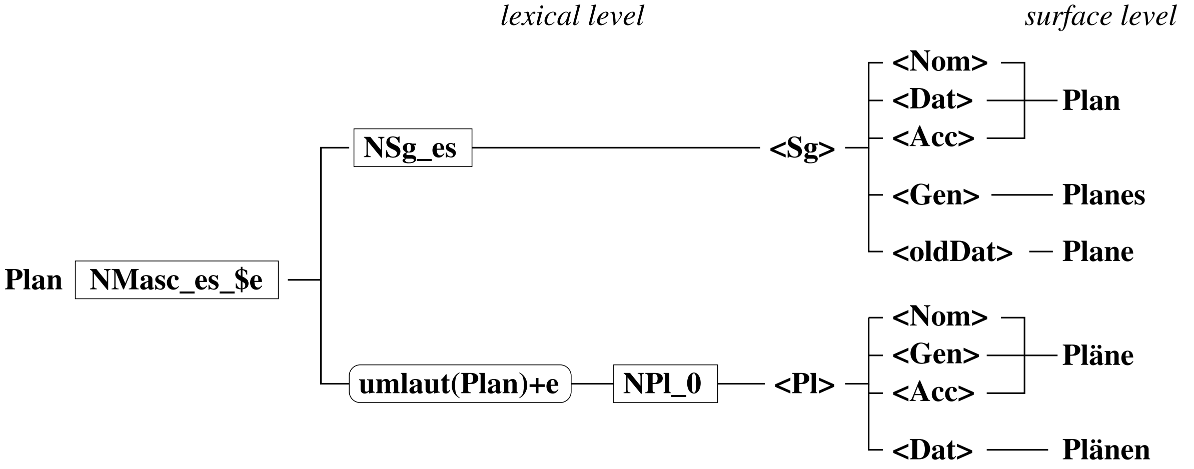


Figure 3.4.: Visualisation of the inflectional variants of *"Plan"* (= "plan") which are implemented in SMOR using the inflection class <NMasc_es_$e>.

native adjective base stem. The grey highlighting straightforwardly visualises the constraint sequences in which stems and affixes must agree in order to be attached: the verb *"heilen"* (= "to cure") can be combined with the suffix *"-bar"* (= "-able") to form the adjective *"heilbar"* (= "curable"), and the noun *"Karte"* (= "card") can be combined with the suffix *"-chen"* (= diminutive German suffix) to the noun *"Kärtchen"*. The same kind of constraints are used for prexation and also for multiple derivations like e.g. Un$_{\textit{Prefix}}$be$_{\textit{Prefix}}$rechen$_{\textit{Stem}}$bar$_{\textit{Suffix}}$keit$_{\textit{Suffix}}$ (= "inability to calculate something").

3.2.4. Inflection

Throughout the previous parts of this section, the tags used to model inflectional classes in SMOR have appeared whenever lexicon entries were shown. Here, we will briefly take a look at the inflection classes, along with an example. The lexicon entry for *"Plan"* (= "plan") is repeated from Table 3.4: <Stem>Plan<NN><base><native><**NMasc_es_$e**>. In Figure 3.4, we list all possible inflectional surface variants for *"Plan"* based on the inflection class *<NMasc_es_$e>*. It can be seen that the main inflection class is first divided into two smaller inflection classes (one for the singular and one for the plural forms) which each contain information on inflectional endings for different grammatical cases of German. In this particular example, an *Umlautung* from *"Plan"* to *"Pläne"* is performed. Internally the *Umlaut* is modeled with a special tag in a two-level rule.

3.3. Previous Approaches

In the mid-1990 years, the interest in developing morphological analysers peaked in the Morpholympics shared task (see Hausser (1996) for an overview). Here, we will take a closer look at **Gertwol**, a system similar to Smor that has widely been used in NLP.

Gertwol is a finite-state morphology system for German which has emerged from the original two-level formalism of Koskenniemi (1983). It has been developed and maintained by Lingsoft[10] and can be obtained through the purchase of a license. Similar systems to Gertwol have been implemented for more than 30 languages.

"Gertwol *was designed to be an industry strength, efficient, wide coverage, general purpose and accurate analyzer/generator which could be used in a wide variety of applications."* (Koskenniemi and Haapalainen, 1996)

Conceptually, Gertwol is very similar to Smor. It consists of a large lexicon of words and morphemes and a set of rules for their combination, both using the two-level formalism of morphological description. For **compounding**, Gertwol makes use of compounding stems which can be combined with base stems for productive word formation. This is identical to compounding in Smor. However, in contrast to Smor, Gertwol does not make use of inflectional classes to produce default compounding stems, but encodes all possible compounding stems of a word directly in the lexicon. While this makes the lexicon much larger, it does not slow down the analysis due to the efficient two-level implementation. Similarly, Gertwol does not allow for productive **derivation**(according to Koskenniemi and Haapalainen (1996)). Instead, derivations are hard coded in the lexicon. This is motivated by two reasons: i) the underlying initial lexicon (the Collins German Dictionary) already contained derivations as full entries and ii) Koskenniemi and Haapalainen (1996) have doubts concerning the productivity of derivation. As for **inflection**, Gertwol relies on inflectional classes, as Smor does. Overall one could say that Gertwol includes more hard-coded information than Smor, which is favourable for analysis precision but might hurt recall whenever unexpected productive phenomena appear.[11]

[10] *Lingsoft*: `http://www2.lingsoft.fi/cgi-bin/gertwol`.

[11] In Appendix A, we give results of an informal comparison of Gertwol and Smor, covering different compounding phenomena.

Gertwol has been widely used in NLP applications: most previous work on compounding in SMT, which made use of a rule-based morphological analyser, rely on Gertwol (e.g., Nießen and Ney, 2000; Popović et al., 2006; Hardmeier et al., 2010).

3.4. Strengths and Weaknesses

After having presented two morphological analysers in detail, this section contains a discussion of the strengths and weaknesses of using rule-based analysers for morphological analysis as opposed to semi- or unsupervised approaches.

Strengths Rule-based morphological analysers provide accurate, linguistically motivated analyses. FST-technology and the two-level model of morphological description makes word processing extremly fast. Morphological analysers allow to handle language-specific phenomena independently of their frequency of occurrence: for example, the fact that the second person singular form of German verbs rarely occurs in German newspaper text has no impact on how such rare forms are analysed. Moreover, integrated rules for word formation enable them to cope with productive processes like compounding or derivation. That means that new compounds created ad hoc are analysed correctly, as long as the compound parts occur in the analyser's lexicon. Finally, even though the efforts of implementation grow with the morphological complexity of the language, rule-based morphological analysers are less complex to implement, as compared to other language-specific NLP tools like e.g. syntactic parsers.

Weaknesses Some of the advantages of rule-based morphological analysers are at the same time disadvantages. In particular, this concerns all language-specific components of the system, as these can usually not be applied to another language. This issue led to the developement of unsupervised approaches to morphological analysis, like e.g. Morfessor, (Creutz and Lagus, 2005).

While we claimed that morphological analysers are less expensive to create as compared to e.g. syntactic parsers, there is still a considerable effort to make, especially with regard to a high-coverage lexicon. Language changes over time, and while morphological analysers can easily cope with new compounds whose parts are contained in their lexicon, they are unable to analyse simple words that are not included in their lexicon (e.g. proper nouns or new technical terms). Therefore, both Smor and Gertwol have been constantly maintained and extended over the last years. In turn, this expensive maintenance period is one of the reasons why many morphological analysers are not freely available. The CLARIN project[12] currently builds a **C**ommon **La**nguage **R**esources and Technology **In**frastructure which collects NLP tools and resources and makes them accessible to researchers via web services. Depending on the purpose of the application (commercial vs. research), the access may or may not be restricted to internal resources (e.g. lexicon, rules).

3.5. Chapter Summary

This chapter contained a discussion of morphological analysers. We introduced the two-level morphology formalism and briefly summarised finite-state technology. Then, we described SMOR in much detail and gave many examples to illustrate how it works, focusing most on compounding. We explained the three different ways of producing compounding stems in SMOR and how they lead to an over-generation of stems.

We also presented GERTWOL, a morphological analyser which is quite similar to SMOR and which has been widely used in NLP applications. Finally, we discussed the strengths and weaknesses of rule-based morphological analysers.

[12] see `http://clarin-d.org/de` (as of July 2014) for more details.

4. Statistical Machine Translation

After having introduced compounds and morphological analysers in the previous chapters, the focus of this chapter is to give some background on statistical machine translation (Smt). Based on statistical models and restricted to the amount of available data, we will investigate the impact of linguistically motivated compound processing on Smt quality later in this thesis .

Origins The idea of statistical machine translation has emerged from the task of deciphering secret messages during the World Wars in the first half of the 20th century. Machine translation considers texts written in a foreign language to be encoded and thus require decoding into a known language in order to make them understandable.

Applications Having emerged from military interests, the application range for machine translation has meanwhile expanded to civil applications. Early approaches in the 1980s focused mainly on the automatic translation of documents from closed domains (e.g. technical documents, instruction manuals). In times of the internet and globalisation, machine translation has found its way into everyday life. Machine translation is applied whenever a human translator is unavailable (e.g. encountering a possibly content-relevant webpage written in a foreign language) or when human translation becomes too expensive (e.g. MT provides pre-translations that will be post-edited by a human translator). As a consequence, the demand for machine translation has been constantly growing.

Structure The remainder of this chapter is structured as follows: we give a general overview of different machine translation approaches in Section 4.1, before we describe statistical machine translation in more detail in Section 4.2. In Section 4.2.3, we discuss two evaluation metrics that have been commonly used in Smt. The chapter concludes in Section 4.3.

4.1. Overview

Before turning to a detailed description of statistical machine translation in the next section, we will introduce approaches to MT in general in this section. Most early approaches to MT were rule-based approaches, working on different levels of analysis (Section 4.1.1). In times of more computational capacities, empirical approaches became more and more popular in recent years. We introduce them in Section 4.1.2. Finally, we discuss the strengths and weaknesses of these two kinds of approaches in Section 4.1.3, where we also briefly discuss hybrid approaches.

4.1.1. Different Analysis Levels of MT

The underlying idea of rule-based machine translation is to define a set of rules to describe the source and the target language and how they are to be translated into oneanother. On the one hand, this collection of linguistic knowledge is a time-consuming and expensive process and it has to be repeated for each new language pair. On the other hand, rule-based approaches usually require fewer computing resources for the translation process. Due to restricted computing facilities, all of the first useful MT systems were rule-based. Hutchins and Somers (1992) distinguish three different rule-based MT approaches, depending on the depth of linguistic analysis involved: *direct translation*, *transfer translation*, and *interlingua translation.* A graphical illustration of these analysis depths, adapted from Vauquois (1968) is given in Figure 4.1.

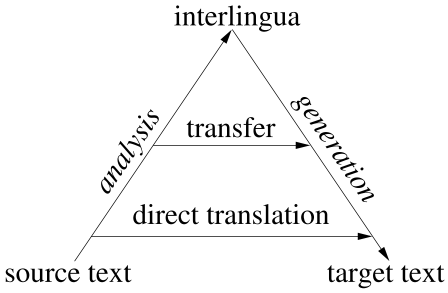


Figure 4.1.: Vauquois' pyramid.

Direct Translation This machine translation design is very language pair and direction specific. It performs at most a morphological analysis of the input sequence, lexicons play a central role. Translation happens at the word (or at most phrase) level, without any major re-ordering possibilities. Thus, the structure of the output is close to the source language structure, but with target language words.

Transfer Translation As can be seen from Vauquois' pyramid in Figure 4.1 above, the transfer translation design contains more analysis and generation components than the direct translation approach. Generally speaking, transfer translations consist of three parts: i) analysis of the input (possibly including a disambiguated syntactic analysis), ii) transfer of this into a target language representation of the same analysis depth and iii) generate fully specified target language text from this representation. In the past, there have been many successful rule-based MT systems that incorporated the transfer translation design.

Interlingua Translation The interlingua approach takes the analysis one step further to an abstract semantic representation, which is universal for (m)any language pairs. The input sequence must be converted into this representation, from which fluent target language output can then be directly generated.

4.1.2. Empirical MT

Empirical approaches to MT build on the assumption that the translation from one language into the other can be learned based on a big textual collection of available translations for this language pair, called *parallel training data* as the texts are parallel at sentence level. In recent years, the growing amount of freely available digital text resources (e.g. via the internet) combined with high-performance computing facilities have led to an increased interest in and benefit of empirical MT approaches.

Example-based MT (EBMT) This MT approach translates by analogy: the text to be translated is broken down into translation units that have occurred in a database extracted from parallel text. The units are translated and then concatenated until they cover the whole input sentence. This approach is well-suited for the translation of complex noun phrases or verb clusters, but there are no means to model agreement across translation units and it does not allow for re-ordering of components.

Statistical MT (SMT) Compared to EBMT, statistical MT goes one step further in that it does not only use translational units from parallel text, but also uses additional statistical models that account for re-ordering of phrases and the fluency of the translation output. In Section 4.2, we describe the general architecture of an SMT system in a more detailed fashion.

4.1.3. Concluding Remarks

Both rule-based and empirical MT approaches, have their advantages and disadvantages. In the following, we summarise general strengths and weaknesses of the rule-based and empirical MT, respectively, and briefly discuss hybrid approaches.

Rule-based MT generates high-quality translations, provided that the required rules and lexicon entries are given. As rule-based MT systems are not dependent on statistics from textual resources, even rare phenomena and exceptions can be translated accurately. In terms of resources, they usually have lower requirements regarding computing facilities for translation. However, the development of rule-based systems is time-consuming as translation-relevant linguistic knowledge must be manually encoded into rules. This is expensive as it requires human experts who ideally should know both languages of the translation pair. In order to cover new terms of a language, rule-based systems require constant maintenance in terms of lexicon updates.

Empirical MT In times of internet and high-performance computing facilities, there is a sufficient amount of resources available to perform empirical MT of acceptable quality. Having access to these computational resources, everyone could theoretically build his own customised SMT system, using freely available parallel data collections (e.g. OPUS[13]) and open source tools to train statistical models (e.g. the MOSES decoder[14]), even without any linguistic background. The drawback of empirical systems is their inability to produce translations of words that have not occured in the parallel training data. Even rare words are challenging, as the statistical models used in translation require a certain amount of data to be reliable.

Hybrid MT The benefits of rule-based and empirical approaches to MT can also be combined in hybrid systems. In general, there are different ways of doing so. For example, the systems can be run in parallel and their outputs can be combined in a post-processing step or frequency statistics can be used to assign linguistic rules with weights.

The morphological compound processing and its application to SMT, which we present in this thesis, can also be considered a hybrid approach, as it combines the benefits of a rule-based analysis with an empirical translation model.

[13] `http://opus.lingfil.uu.se/`, as of July 2014.
[14] `http://www.statmt.org/moses`, as of July 2014.

4.2. Statistical Machine Translation (SMT)

The underlying idea of SMT is to learn translations based on statistical models. These are trained on a multilingual text collection that has been parallelised at sentence level beforehand. Traditional SMT systems usually only translate words that have occured in the parallel text they have been trained on – a shortcoming that our compound-processing approach alleviates. We will motivate compound processing for SMT in Chapter 5. The present section gives a general overview of statistical machine translation.

Terminology In recent years, different kinds of SMT systems have emerged. They can be classified according to the granularity of their translational units. Initial systems operating on word-level translations are referred to as **word-based** SMT. Today, many state-of-the-art systems make use of translations at the phrase level, thus called **phrase-based** SMT. Note that phrases are sequences of words and need not correspond to linguistic units. The usage of phrases instead of words has certain advantages. On the one hand, it facilitates the translation of constructs that have differing granularities in the source and target language. On the other hand, it improves target language fluency, as local syntactic agreements can be passed through (e.g. German adjective-noun pairs). More recently, **tree-based** SMT systems emerged, which make use of syntactic structures for translation unit selection. The latter two approaches have also been combined in **Hierarchical phrase-based** SMT systems, where the advantages of phrase- and tree-based SMT are combined. For an overview of these different approaches, see (Lopez, 2008; Koehn, 2009).

The SMT system we use throughout this thesis is a phrase-based SMT system (PBSMT). However, for the sake of simplicity and as we do not contrast this approach with other approaches within this thesis, we simply refer to it as SMT henceforth.

System Architecture The translation process of an SMT system can be described using the Noisy-Channel Model (Shannon, 2001). The underlying idea of the Noisy-Channel Model is the following: assume we have a signal or message e that is transmitted through a noisy channel. The outcome is no longer identifiable as the original message, but a noisified version of it, namely *f*. In SMT, the input message e corresponds to the source language input and the output message f corresponds to the target language output. The task of SMT is to find an appropriate decoding of the output f which maps

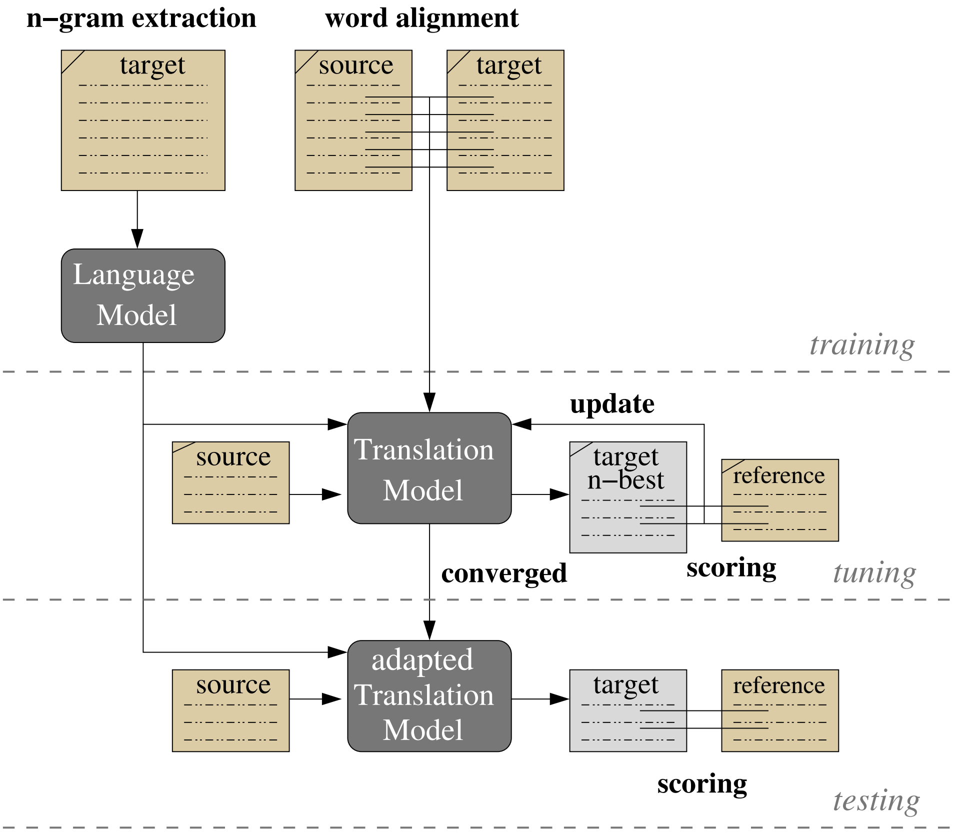


Figure 4.2.: Graphic illustration of a standard SMT architecture.

it to the input *e*. This is done using two kinds of models: i) a **language** model on the target language, which accounts for the fluency of the output and ii) a **translation** model that determines the probability of a source language being translated into a target language.

In practice, applying an SMT system usually consists of three steps: training a language model and a translation model, tuning translation weights and testing the translation quality. See Figure 4.2 for an illustration of the general architecture. Despite their differing levels of granularity, all SMT variants introduced in the previous paragraph share this general structure.

Tools For the SMT experiments in this thesis, we use Moses (Koehn et al., 2007), an open source toolkit to build phrase-based statistical machine translation systems. It can be obtained from `http://www.statmt.org/moses/` (as of July 2014), together with a detailed documentation of the components and their implementations. The Moses toolkit

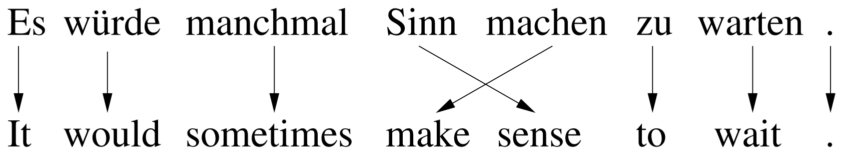

Es würde manchmal Sinn machen zu warten .
It would sometimes make sense to wait .

Figure 4.3.: Illustration of word alignment from German to English.

also includes Giza++ (Och and Ney (2003), Gao and Vogel (2008)), which is commonly used for statistical word alignments.

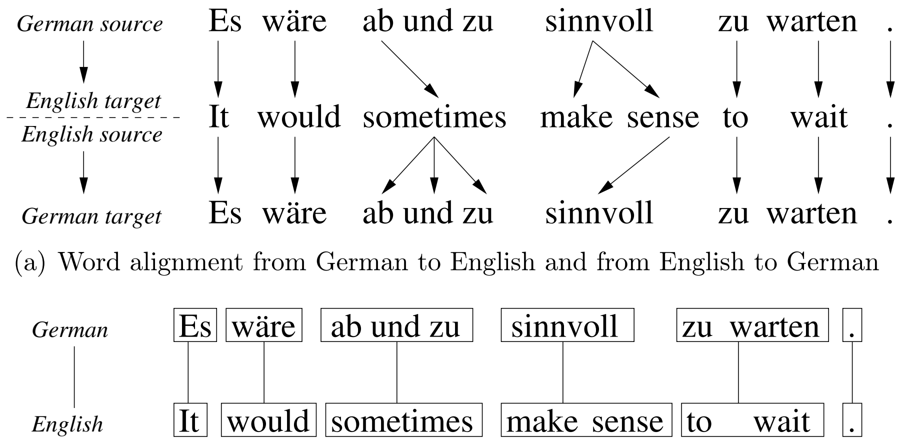
Structure The remainder of this section describes each of the general steps of a standard SMT system: training, tuning and testing, respectively. Following the architecture given in Figure 4.2, the focus of Section 4.2.1 is on the training step: it describes statistical word alignment on parallel training data, which is the basis for the translation model. Training also includes a language model trained on target language data. This model will account for translation fluency. Section 4.2.2 deals with the tuning of feature weights in the translation model and in Section 4.2.3, we describe two commonly used metrics for the evaluation of SMT systems on held-out testing data.

4.2.1. Model Training

The mathematical details of statistical machine translation are beyond the scope of this thesis. The interested reader is referred to Koehn (2009) for a detailed introduction to the models, including their mathematical background. The description here is limited to a comprehensive summary of the system components. It covers word alignment, building a phrase table and a reordering model (both based on the word alignment), and finally training a language model. It concludes with a description of how these components are integrated into one translation model.

Word Alignment Given a bilingual, sentence-aligned text, the task of word alignment is to find translational correspondences of source language words in the target language. See Figure 4.3 for a word alignment example from German to English.

In the 1990s, Brown et al. (1992) from IBM introduced five models to compute word alignments with different levels of complexity. These have become established components of SMT and have been widely used ever since. We briefly describe them in ascending order of complexity, with all preceding models being included in the current model: **IBM**


German source
Es wäre ab und zu sinnvoll zu warten .
English target
It would sometimes make sense to wait .
English source
German target
Es wäre ab und zu sinnvoll zu warten .
German
Es | wäre | ab und zu | sinnvoll | zu warten | .
English
It | would | sometimes | make sense | to wait | .

(a) Word alignment from German to English and from English to German

(b) Bidirectional phrase alignment.

Figure 4.4.: Visualisation of phrase extraction based on symmetrised word alignments.

Model 1 accounts for a simple lexical translation probability of a source word being translated to a target word, based on the number of parallel sentences in which the word pair has occured. Note that the position of the words within the sentence has no impact on this probability. In order to increase the probabilities for correct word order, **IBM Model 2** takes the positions of the source and target words into consideration as well, i.e. how often a word at position x of the source sentence has been translated to a word at position y of the target sentence. In addition to that, **IBM Model 3** models the fertility of a source word, i.e. the probability of the source word being translated into one or more target words (or none!). This model even allows the insertion of target words that have no matching counterpart in the source language. **IBM Model 4** has an improved, *relative* model of distortion, which makes the position of the current target word dependent on the position of the translation of the source word. Finally, **IBM Model 5** removes some deficiencies: impossible alignments (e.g. two words at the same position of the target sentence) are removed. For more details on all of these models see Koehn (2009).

The probability estimates of the IBM Models are obtained from applying **expectation-maximisation** (EM) algorithm (Dempster et al., 1977). First, all model parameters are initialised with uniform probabilities. The EM algorithm consists of two steps that are iteratively repeated until convergence: the expectation step assigns probabilities to be expected based on the current parameters of the model. The maximisation step computes new parameters in order to maximise this expectation.

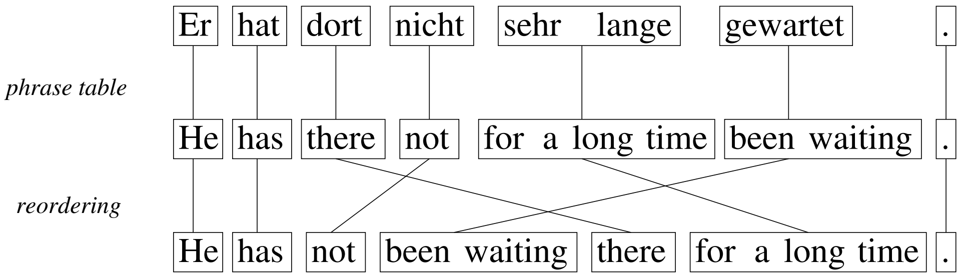


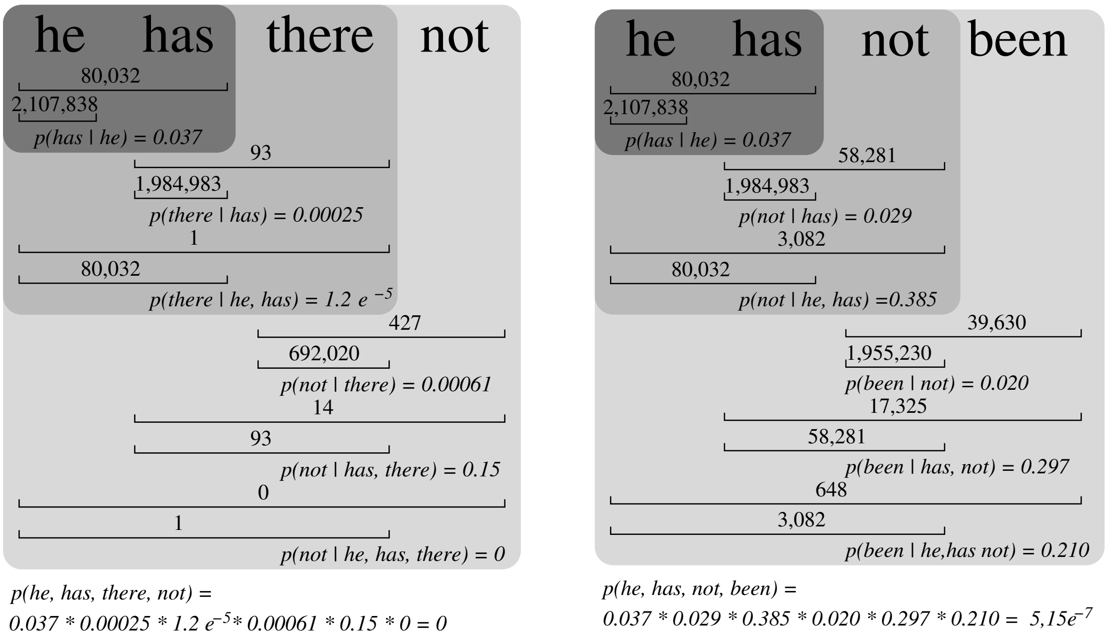
Figure 4.5.: Visualisation of phrase reordering.

Throughout this thesis, we use the freely available multithreaded GIZA++ toolkit (Och and Ney (2003), Gao and Vogel (2008)) to estimate word alignments for the SMT systems we build.

Phrase Table The performance of the five IBM models alone is comparable to that of a word-based SMT system. They allow for **1:n** alignments from source to target language, but a target word can never be aligned to more than one source word. An example is given in Figure 4.4 (a): the word alignment from German to English allows *"sinnvoll"* to be aligned to "makes sense", but *"ab und zu"* cannot be aligned to "sometimes". In order to overcome this issue and to allow for **m:n** alignments (and thus phrases), some additional processing is required: first, the word alignment requires symmetrisation, i.e. it is to be run in both alignment directions (source to target and vice versa) and the resulting two alignments are to be intersected. Then, additional alignment points are *grown* (= "selected") from the unification of the two word alignments. This is an important step that distinguishes pure word alignment from phrase-based SMT.

A phrase pair is considered to be valid if none of the words covered by the pair is aligned to a word outside the phrase pair. In Figure 4.4 (b), this is the case for the phrases sinnvoll-makes sense, ab und zu-sometimes and zu warten-to wait. Note that SMT phrases must not necessarily correspond to linguistic units (e.g. complex adverbs, verbal clusters, noun phrases) and theoretically, there is no limit to the maximal phrase size. However, there is a natural trade-off between phrase lenght and frequency of occurrence.

Having collected valid phrase pairs from the symmetrised word alignment, each phrase pair is assigned a probability. It is based on the frequency of occurrence of the phrase pair and the frequency of independent occurrences of the source phrase. All phrases and probabilities are stored in the phrase table.



(a) Disfluent English word order:
no support from the language model.

(b) Fluent English word order:
language model supports this option.

Figure 4.6.: Visualised calculation of language model scores for the original (a) and reordered (b) word order of the example from Figure 4.5. Up to 4 preceding words and possible subsequences thereof are considered. For example, for the probability of generating "has", after "he", take the frequency of the pair "he has" (80,032) divided by the frequency of "he" (2,107,838), which equals 0.037.

Reordering Model In order to account for differences in word order between the source and the target language, it is often not sufficient to reorder single words (as it happens in word alignment). Instead, whole phrases must be moved to achieve acceptable target language fluency. An example is given in Figure 4.5. The reordering model usually consists of a penalty score that increases with the distance between the original position of the phrase and its reordered position. The penalised reordering option thus only applies in cases where there is substantial support from the language model. In the following paragraph on language modeling, we will show that this assuption holds for the given example.

Language Model In contrast to the other models presented so far, the language model does not depend on the source language. Instead, it is trained exclusively on target language data. The purpose of a language model is to improve the fluency of the SMT target language output. It is usually based on n-gram frequencies extracted from a corpus. In the course of translation, the generation probability of a current word is (amongst others) made dependent on the sequence of words that have already been generated. The language model assigns weights to this sequence depending on n-gram frequencies of the sequence with and without the word to be generated. In Figure 4.6, we give a simplified illustration of how the language model is applied. N-gram spans are illustrated through horizontal bars, with the frequencies of the n-grams on top of them. The respective conditional probability scores are given beneath each sequence pair (with/without word to be generated). In this example, a reordering (see previous paragraph) from *"he has there not for a long time been waiting."* into *"he has not been waiting there for a long time."* is indicated, as the original word order of the output is not supported by the language model. Note, however, that the probabilities are usually smoothed during language model training in order to avoid zero probabilities like the one given in Figure 4.6. Smoothing assigns non-zero probabilities even to rare or unseen n-grams.

Translation Model Finally, the probabilities from the phrase table, the reordering model and the language model are combined into a *log-linear model* in which the translation probability is composed of a set of weighted feature functions. In SMT, these feature functions correspond to the phrase table, the reordering model and the language model. The weights have a uniform distribution by default. See Koehn (2009) for mathematical details. During tuning, the weights will be adjusted according to their importance for the translation performance (cf. Section 4.2.2).

4.2.2. Tuning of Feature Weights

We already introduced the translation model in the previous paragraph: it consists of several different models whose probabilities are combined within a log-linear model. By default, uniform weights are assigned to each of these models. In this section we describe how these weights can be tuned for an optimal translation performance. In Figure 4.7, we repeat the tuning segment from Figure 4.2 above. Tuning in SMT is usually performed with minimum error rate training (MERT, Och, 2003).

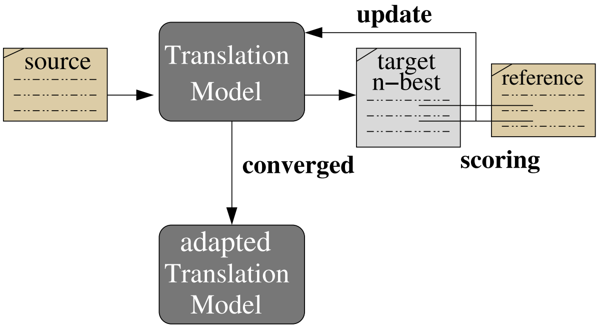


Figure 4.7.: Iterative tuning of feature weights with minimum error rate training. A source language text is translated with current feature weights. The outcome (in form of n-best translations) is scored against a human translation reference. The weights are then updated accordingly for the next iteration, until convergence.

It is an iterative process, where each iteration includes:

i) translate a source language text with current feature weights
ii) score the outcome against a human reference translation
iii) update feature weights accordingly
iv) repeat iteration until convergence

The source language text segment must not be part of the parallel training data for the translation model. It usually consists of 1,000 - 3,000 sentences and requires a human reference translation. The n-best translations (often with n=100) for each sentence are scored against this reference translation using Bleu, a common evaluation metric that will be introduced in Section 4.2.3 below. The weights for the different models are then adjusted accordingly for the next iteration. This process is repeated until convergence. The final weights are stored and used for the translation of the test set, which should itself be disjoint from the training data and the tuning set.

4.2.3. Evaluation Methodology

The evaluation of translation quality is a challenging task. It is often difficult to clearly draw a line between where a correct translation ends and false one begins. Moreover, due to the variety of natural language, there is usually more than one correct translation of a given source language text. This applies to any language pair.

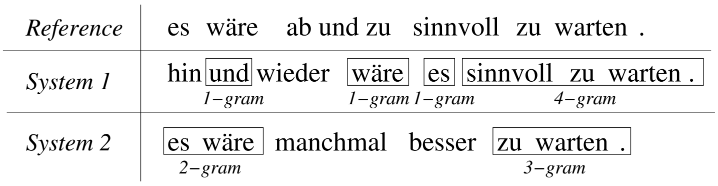
In the following, we will first discuss human evaluation and then present two automatic evaluation metrics, that are widely used to compare SMT systems, namely BLEU (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007).

Human Evaluation The idea of human evaluation is to let humans manually judge translation quality. For this, it is desirable to ask native speakers of the target language, who, ideally, also know the source language of the translation task to be evaluated. This allows the human annotator to judge the translations by taking the source language text into consideration. In cases where the human annotator is not familiar with the source language, one can alternatively use a human reference translation to which the translation under consideration is then compared.

Human annotators are often requested to judge translation quality with respect its *adequacy* and *fluency*. In order to facilitate the judgements, i.e. to allow for more than just *correct* or *false* translations, previous work proposes graded scales for adequacy (5 = flawless English, 4 = good, 3 = non-native, 2 = disfluent, 1 = incomprehensible) and fluency (5 = all meaning, 4 = most, 3 = much, 2 = little, 1 = none), respectively (see, e.g., Koehn (2009) for more details and pointers to previous work). However, there is often only little agreement between the human judges. It has thus become common practise (e.g. in the evaluation of the annual WMT shared tasks) to compare the outcome of several systems on the sentence level and to let human annotators rank them according to preference.

Low agreement between human judges is only one problem of human evaluation. It is expensive and time-consuming and suitable human annotators may not be available. Moreover, the judgements are not reproducible and due to the subjectivity of the decision, the results of different systems (e.g. of different research groups with different human judges) are hard to compare.

Automatic Evaluation Automatic evaluation methods rely on the assumption that *"the closer a machine translation is to a professional human translation, the better it is"*. (Papineni et al. (2002), p. 311). They are usually conceived to calculate the distance to one (or more) human reference translation(s). One the one hand, automatic MT evaluation methods are strictly speaking not evaluating the quality of the translation (as they do not take the source language into account), but the similarity to one (or possibly several) correct reference translation. On the other hand, the advantages of


Reference
es wäre ab und zu sinnvoll zu warten .
System 1
hin und wieder wäre es sinnvoll zu warten .
1−gram 1−gram 1−gram 4−gram
System 2
es wäre manchmal besser zu warten .
2−gram 3−gram

(a) Reference translation with translation outputs.

	System 1	System 2
1-gram precision	7/9	5/9
2-gram precision	3/8	3/8
3-gram precision	2/7	1/7
4-gram precision	1/6	0/6
brevity penalty	9/9	7/9
Bleu-1	77.78%	43.21%
Bleu-2	29.16%	16.20%
Bleu-3	8.33%	2.31%
Bleu-4	1.39%	0%

(b) Calculation of Bleu scores based on n-gram matches.

Figure 4.8.: Example for Bleu score calculations, adapted from Koehn (2009), p. 257ff. In (b), the counter denotes the number of matching n-grams, the denominator indicates the number of n-grams to be found. The brevity penalty reflects the number of words produced with respect to the number of reference words, irrespective of their correctness. Bleu-2 combines all precisions up to 2-grams (e.g. 7/9 * 3/8 * 9/9 = 29.16%), Bleu-3 up to 3-grams, etc.

using automatic methods are obvious: they are always available, cheap and fast, and their results are reproducable and comparable to other systems (provided that the same data were used). Furthermore, automatic methods can be used to tune the translation system performance with respect to some developement set (without overlap to the training or testing data).

Bleu The **B**i**L**ingual **E**valuation **U**nderstudy (henceforth: Bleu), introduced by Papineni et al. (2002) performs an exact character matching of word n-grams (typically up to 4-grams) against one (or more) human reference translation(s). In order to account for missing words, Bleu incorporates a brevity penalty for each word that the translation output is shorter than the reference translation. In Figure 4.8, we give a detailed example of how Bleu is calculated. The n-gram precisions reflect how many of the n-grams produced by the translation system exactly match n-grams of the reference translation. The brevity penalty indicates the number of words produced with respect to the number of words of the reference translation, irrespective if they match the reference or not. Finally, each of the n-gram precisions is combined with the brevity penalty to get Bleu scores. For example, 2-gram Bleu is calculated by combining the 1-gram precision with the 2-gram precision and the brevitiy penalty. Furthermore, it can be seen from Figure 4.8 (Bleu-4, system 2 output) that the Bleu-score is 0 as soon as any of the n-gram precisions is 0. In order to avoid zero Bleu scores, they are usually calculated on the document level.

The Bleu evaluation is freely available,[15] simple to use and runs fast. Finally, Papineni et al. (2002) could show that Bleu correlates well with human judgements of translation quality. Despite some critisism over the years, Bleu is currently the most popular automatic evaluation metric for Smt.

Meteor While the Bleu score introduced above is a purely precision-oriented metric, we will here present Meteor (**M**etric for **E**valuation of **T**ranslation with **E**xplicit **OR**dering, Lavie and Agarwal, 2007), a metric which also takes the recall of a translation into account. This is important, as recall reflects the amount of meaning that has been covered by the translation. In contrast to Bleu, where fluency is modeled using n-grams, Meteor takes only unigrams into consideration and uses the number of chunks (containing adjacent words in both the source and the target language text) to penalise disfluent output. Unigram matching is performed through establishing a word alignment between the system output and the reference translation. First, exact matching unigrams are counted, then, the model backoffs to i) a stemmed version of the unigram and ii) semantically equivalent representations of the unigram, derived from (WordNet, Fellbaum, 1998). This procedure overcomes two shortcomings of Bleu, which neither credits translations with the correct lexeme when it differs with respect to inflection or POS, nor does it credit the usage of synonyms that convey the same meaning as the respective word used in the reference translation.

However, this potential goes at the expense of computing complexity and speed. Moreover, even though (as of 2014) Meteor comes with stemmers for a considerable number of languages, semantic class hiearchies like the English WordNet are expensive to create and thus hard to access for languages other than English. Nevertheless Meteor can be used for any target language. In lack of a stemmer and/or a semantic class hierachy, only the exact matching method will then be performed.

[15] `ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13.pl`, as of July 2014.

4.3. Chapter Summary

This chapter dealt with machine translation. It covered some general concepts, including rule-based and empirical approaches. We first presented some different analysis (and generation) depths according to the pyramid of Vauquois (1968). Then, we discussed the strengths and weaknesses of rule-based and empirical approaches, before we presented statistical machine translation (Smt) in more detail. Smt can generally be divided into the three parts, *training*, *tuning* and *evaluation*, each of which we presented in detail and along with illustrative examples. At this point of the thesis, we have introduced both the productive word formation in German and the fundamentals of Smt. In the next chapter, we will motivate the usefulness of compound processing for Smt.

5. Compound Processing in SMT

In the previous chapters, we have introduced German compounds, morphological analysers and basic concepts of Statistical Machine Translation (Smt). In this chapter, we will now motivate the usefulness of compound processing in Smt in detail. We discuss the compositionality of compounds which is a crucial assumption for the approach to work. Moreover, we show possible locations of compound processing in SMT.

Structure The remainder of this chapter is structured as follows: we first motivate the general idea behind compound processing in Section 5.1, including its restriction to best work on compositional compounds. Then, we locate the compound processing we perform within a classical SMT pipeline and also mention alternative approaches in Section 5.2. In Section 5.3, we briefly discuss evaluation matters before the chapter concludes in Section 5.4.

5.1. Motivation

Productive processes like compounding are challenging for statistical machine translation, as they lead to a high number of non-translatable words missing in the training data. As the amount of parallel training data is usually limited, it is desirable to make the most out of the available data. While many compounds do not have occurred in the training data, most of their parts usually have occurred. For example, in the course of experiments with speech recognition systems, Berton et al. (1996) found that for 54.4% of the compounds, all component words were listed in the lexicon; for 36.7%, at least one component word has occurred and only for 8.9% of the compounds, none of the component words was covered by the lexicon (cf. Berton et al. (1996), p. 1166).

Even though Smt is based on large training corpora instead of a lexicon, these findings are transferable to the data sparsity problem of Smt. Splitting compounds gives the Smt model access to the simple component words and thus makes many compounds translatable part-by-part. In the following, we discuss compositionality (Section 5.1.1), which is a crucial factor for the approach we pursue. Then, we first motivate compound splitting (Section 5.1.2) and then compound merging (Section 5.1.3).

5.1.1. Compositionality

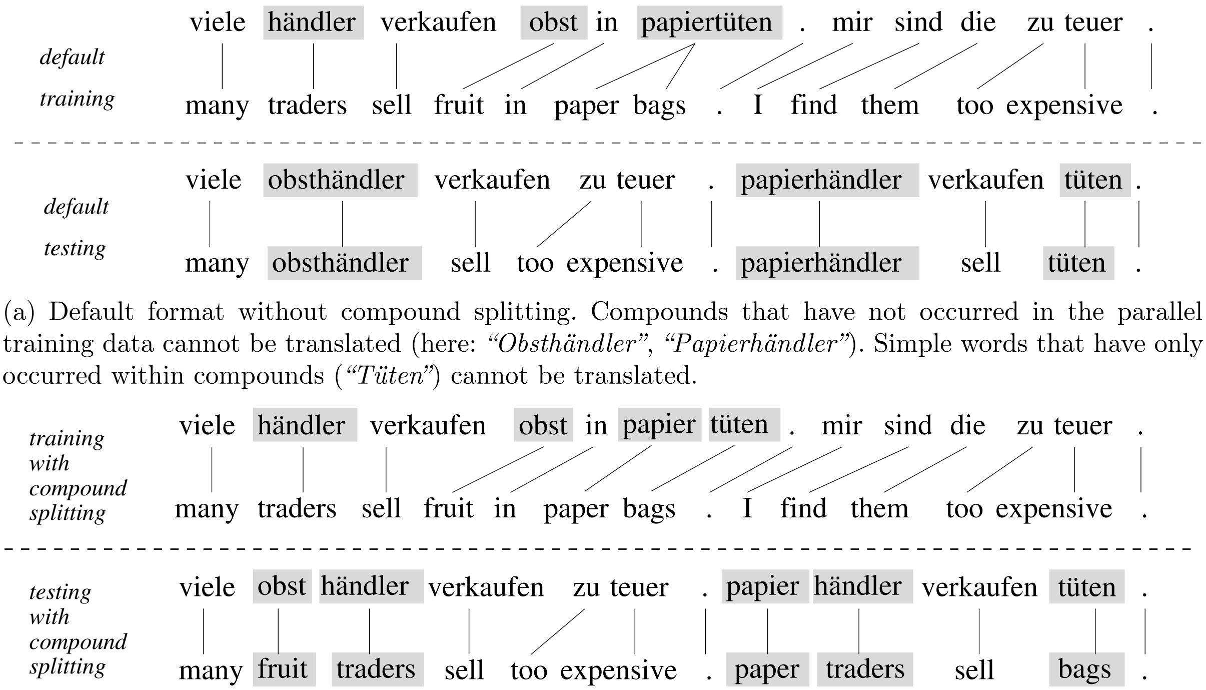
The compound processing approach we present in this thesis is conceived for compositional compounds. In fact, most German compounds have a compositional semantics, i.e. the meaning of the compound can be derived from the meaning of its parts: e.g. *"Papier+Tüte = Papiertüte"* (= "paper + bag = paper bag") is a bag made of paper. However, German also features semantically opaque compounds, which have either a less transparent meaning, e.g. *"Wunder+Tüte = Wundertüte"* (= "wonder + bag = grab bag"), where the bag contains toys and/or sweets instead of a wonder, but still, it denotes a bag, or they are completely opaque, like *"Tran+Tüte = Trantüte"* (= "fish oil + bag = s.o. being lame").[16]

So far, we do not explicitly differentiate between transparent vs. opaque compounds in our compound processing approach, even though the latter ones should be prevented from being split. Note however, that opaque constructions are hardly productive, while semantically transparent compounds are highly productive. German opaque compounds are usually lexicalised and should thus more probably have occured (more frequently) in the training data than newly created compositional compounds generated from scratch.

Nevertheless, both in the course of compound splitting and merging opaque compounds are implicitly addressed. For example, we restrict our compound splitting to compounds which are not lexicalised, i.e. they have not occurred as a whole in the lexicon of the rule-based morphological analyser (See the paragraph on *filter flags* in Section 7.2.1, on page 86 below for more details). In the course of compound merging, we use a feature for two German output words that have been aligned from one English word. This indicates compounds that should not have been split in the first place, either because they are lexicalised in English or because the German compound has an opaque semantics (see also paragraph *alignment-based features* in Section 11.3.2, on page 158).

In the future, we plan to more explicitly address the compositionality of compounds in the course of compound splitting.

[16] One possible explanation for the origin of this opaque compound is the viscous (= slowly flowing) consistency of the fish oil gained from e.g. whales in the 19th century for the fabrication of soaps .


default training
viele händler verkaufen obst in papiertüten . mir sind die zu teuer .
many traders sell fruit in paper bags . I find them too expensive .
default testing
viele obsthändler verkaufen zu teuer . papierhändler verkaufen tüten .
many obsthändler sell too expensive . papierhändler sell tüten .
training with compound splitting
viele händler verkaufen obst in papier tüten . mir sind die zu teuer .
many traders sell fruit in paper bags . I find them too expensive .
testing with compound splitting
viele obst händler verkaufen zu teuer . papier händler verkaufen tüten .
many fruit traders sell too expensive . paper traders sell bags .

(a) Default format without compound splitting. Compounds that have not occurred in the parallel training data cannot be translated (here: *"Obsthändler"*, *"Papierhändler"*). Simple words that have only occurred within compounds (*"Tüten"*) cannot be translated.

(b) Compound splitting gives the SMT model access to the component words. Compounds that have not occurred in the parallel training data (here: *"Obsthändler"*, *"Papierhändler"*) can thus be translated part-by-part. Moreover, even simple words, that have only occurred within compounds (here: *"Tüten"*) can now be translated.

Figure 5.1.: Illustrative example of how compound splitting enables the translation of compound that have not occured in the parallel training data. Phrase boundaries are omitted to enhance readability.

5.1.2. Compound Splitting

As stated earlier, compounds which have not occurred in the parallel training data, cannot be translated. When translating from German to English, compound splitting enables the translation of compounds part-by-part, provided that the component words of the compound have occurred in the training data. An example is given in Figure 5.1. For the sake of simplicity of this example, assume that the two sentences given in the parallel data section are all parallel training data available. The compound *"Obsthändler"* (= "fruit traders") has not occurred in that parallel training data and can thus not be translated. Instead, it is transferred as it is, cf. Figure 5.1(a). However, it can be seen from the training data, that both *"Obst"* (= "fruit") and *"Händler"* (= "traders") have occurred in the data. Splitting compounds prior to translation, allows to access these translations and thus to translate the compound part-by-part cf. Figure 5.1(b).

Different granularities of the source and target language often lead to 1:n word alignments, which are less reliable than 1:1 alignments. Splitting compounds prior to training transforms 1:n alignments of compositional German compounds, where each component word of the compound corresponds to a separate English word, into 1:1 alignments, cf. *"Papiertüten"* (= "paper bags") in the training data sections of Figures 5.1(a)+(b).

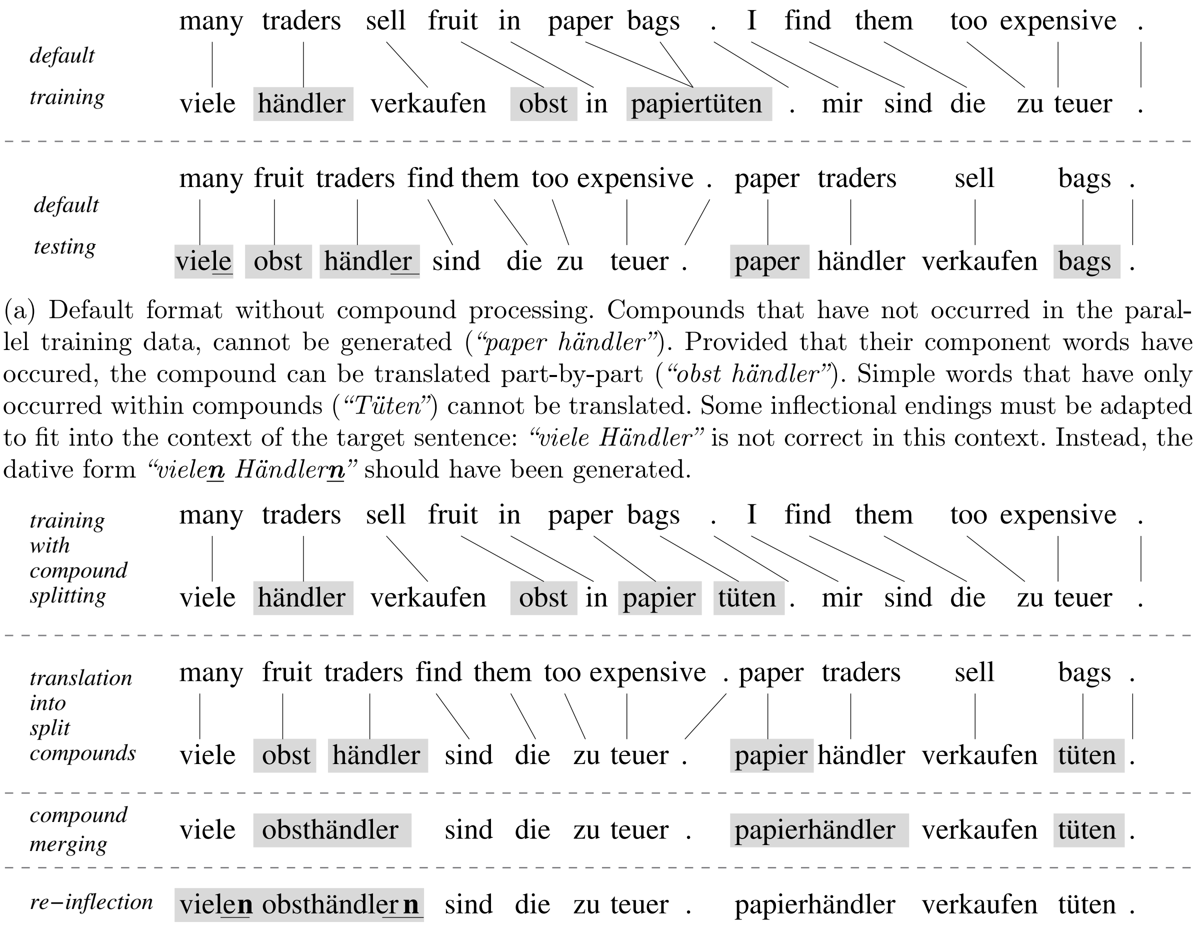
Moreover, this procedure also allows the translation of compounds, with component words that have only occurred within other compounds in the training data, e.g. *"Papierhändler"* (= "paper traders") of Figure 5.1, whose modifier only occurred in the compound *"Papiertüten"* (= "paper bags") whereas the head occurred as a separate word, *"Händler"* (= "traders"). The same applies for simple words that have solely occurred within compounds in the parallel training data, e.g. *"Tüten"* (= "bags"). Finally, the splitting of compounds prior to training increases the frequency counts of simple words, which in turn enhances the probability for a correct translation.

5.1.3. Compound Merging

In general, one can say that the issues observed above for the German to English translation direction remain in the opposite direction, namely English to German. Again, words that have not occurred in the training data, cannot be translated. However, in this direction, the unknown closed compounds occur on the target side.

Consider for example Figure 5.2 (a), where the default system is not able to generate the correct German translation *"Obsthändler"* based on the two English input words "fruit trader". Even though the default system has no compound processing, it can still translate the two English words into two German words (namely *"Obst Händler"*), as these have occurred as separate words in the training data. Nevertheless, the default system cannot merge the two words into the compound *"Obsthändler"*. Moreover, a closer look at the target sentence reveals that the inflection of *"viele"* (= "many") and *"Händler"* (= "traders") is not coherent in this context. In order to make it a sound German sentence, these two components must bear dative inflection, namely *"viele**n** Händler**n**"*. Furthermore, the translation result of the default system in Figure 5.2 (a) shows that simple words which previously only have occurred in closed class compounds cannot be translated (e.g. "paper" - *"Papier"* and "bags" - *"Tüten"*).

In Figure 5.2 (b), we illustrate how compound processing can overcome these shortcomings. Splitting compounds prior to SMT training enables the translation of words


default training
many traders sell fruit in paper bags . I find them too expensive .
viele händler verkaufen obst in papiertüten . mir sind die zu teuer .
default testing
many fruit traders find them too expensive . paper traders sell bags .
viele obst händler sind die zu teuer . paper händler verkaufen bags .
training with compound splitting
many traders sell fruit in paper bags . I find them too expensive .
viele händler verkaufen obst in papier tüten . mir sind die zu teuer .
translation into split compounds
many fruit traders find them too expensive . paper traders sell bags .
viele obst händler sind die zu teuer . papier händler verkaufen tüten .
compound merging
viele obsthändler sind die zu teuer . papierhändler verkaufen tüten .
re−inflection
vielen obsthändlern sind die zu teuer . papierhändler verkaufen tüten .

(a) Default format without compound processing. Compounds that have not occurred in the parallel training data, cannot be generated (*"paper händler"*). Provided that their component words have occured, the compound can be translated part-by-part (*"obst händler"*). Simple words that have only occurred within compounds (*"Tüten"*) cannot be translated. Some inflectional endings must be adapted to fit into the context of the target sentence: *"viele Händler"* is not correct in this context. Instead, the dative form *"viele**n** Händler**n**"* should have been generated.

(b) Compound splitting prior to training gives the SMT model access to the component words. This enables even part-by-part translation of compounds whose component words only have occurred within other compounds in the training data (*"papier händler"*). Moreover, separate occurrences of former component words can be translated (*"tüten"*). In a post-processing step, compounds are merged (*"Obsthändler"*, *"Papierhändler"*) and suitable inflectional endings are generated (*"viele**n** Obsthändler**n**").*

Figure 5.2.: Illustrative example on how compound processing, when combined with inflection prediction in the English to German translation direction allows for the generation of compounds that have not occurred in the parallel training data, and unseen inflectional variants thereof. For the sake of simplicity, we omit phrase boundaries.

that only have occurred within compounds in the original training data (e.g. *"Papier"* and *"Tüten"*). Moreover, splitting also increases the frequency counts of simple and thus their translation probability. After translation, a two-step postprocessing is performed in which compounds are first (re-)merged and then the inflection of the whole sentence is adjusted, if necessary.

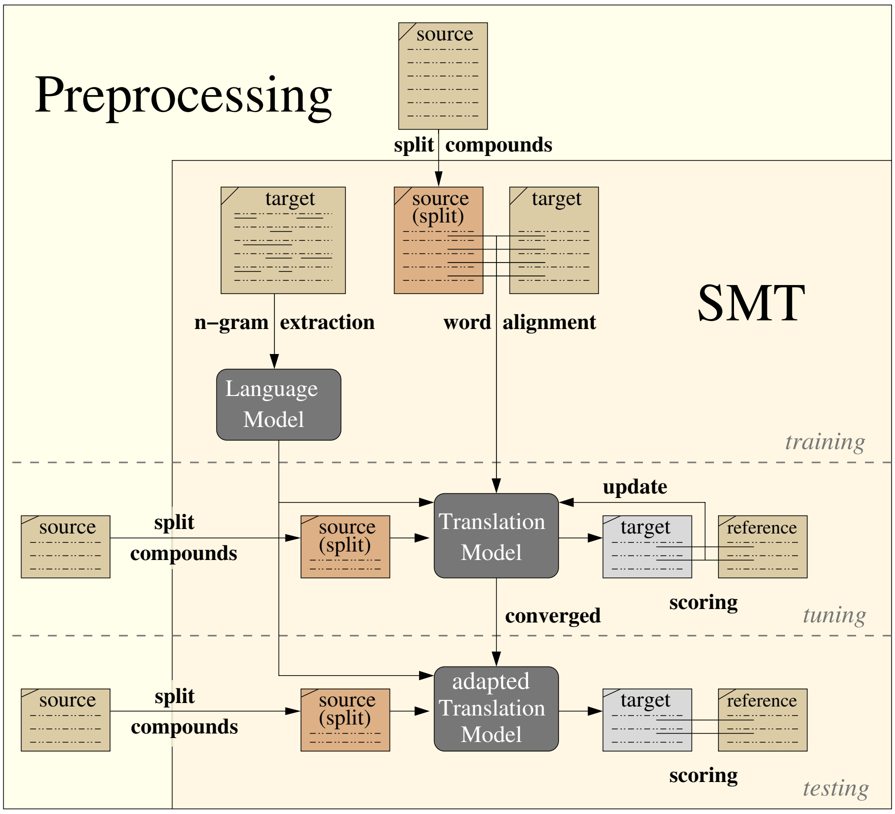


Figure 5.3.: German to English translation: compound splitting is performed as a preprocessing step, prior to SMT training. Besides the compounds in the source language sections of the parallel training data, even the compounds in the tuning data and testing data must be split prior to translation.

5.2. Integration of Compound Processing in Smt

In this section, we locate compound processing within a classical SMT pipeline. We first present a simple pre-/postprocessing approach, which we adapted for the integration of compound processing into SMT in this thesis (Section 5.2.1). Then, we give an overview of alternative approaches for compound processing in SMT (Section 5.2.2).

5.2.1. Pre-/Postprocessing Approach

A simple way to integrate compound processing into SMT is to modify the data on which the SMT system will be trained. Compound splitting is applied prior to training, irrespective if German is the source or the target language. The SMT models are then

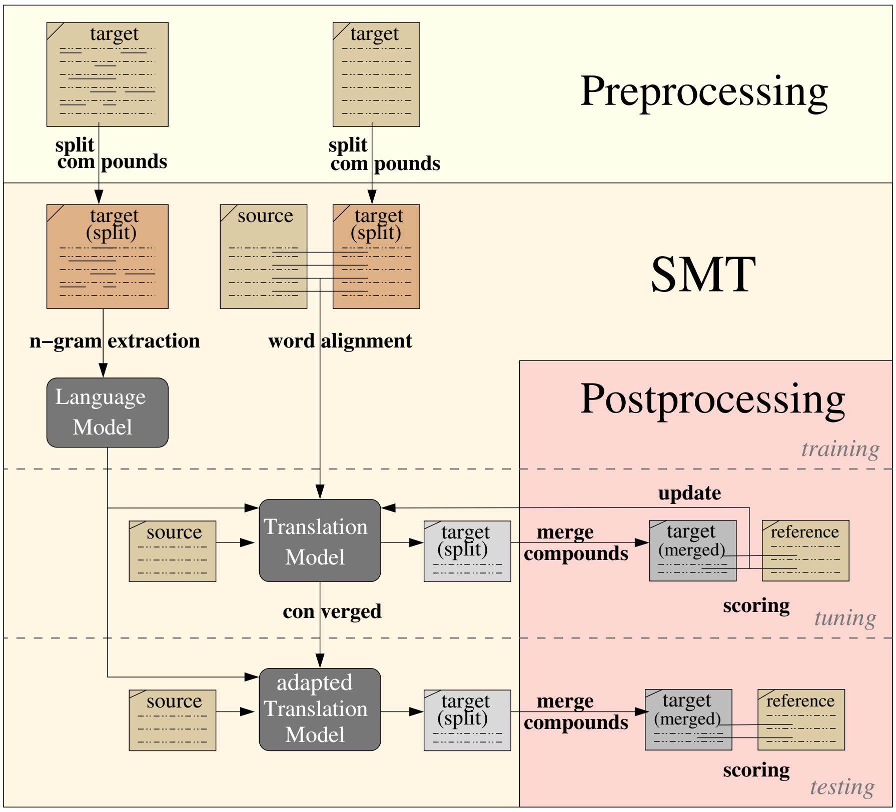


Figure 5.4.: English to German translation: in a pre-processing step, the compounds of the German target language data are split, prior to SMT training. After translation into this split representation, a post-processing step is required to merge simple words into compounds. Note that tuning is performed against a reference translation in which compounds are **not** split. In each iteration of MERT, we thus merge simple words of the output into compounds and thereby integrate compound merging into the scoring process.

trained on original English and a split representation of the German data. If German is the source language of the system, even the compounds of the tuning and testing sets must be split before translation. An illustration of this preprocessing, for the German to English translation direction is given in Figure 5.3. Note that the general SMT architecture remains unchanged (cf. Figure 4.2 on page 40).

In the opposite translation direction, from English to German, a combination of pre- and postprocessing is required for an appropriate compound processing. Again, compounds are split prior to training and the SMT models are trained on original English and a split representation of the German data. However, after translation, compounds must be re-merged before the output can be scored against a human reference translation. An illustration of this combined pre- and postprocessing is given in Figure 5.4. Note that tuning is performed against a human reference translation in which compounds are not split. In each iteration of MERT, we thus merge simple words of the output into compounds. This way, the quality of the compound mergings (in terms of number of words) is implicitly scored. At testing time, the compounds of the German output must be merged, before the output is scored against a human reference translation.

Inflection Prediction Not only compound processing, but also inflection prediction is performed within a pre- and a postprocessing step. For the sake of simplicity of the illustration, the example given in Figure 5.4 does not include the lemmatisation and re-inflection component. In the real pipeline, compounds are split and lemmatised prior to training and in tuning the output is scored against a lemmatised version of the human reference translation in which compounds are not split. After translation of the testset, compounds are first merged and then inflection is predicted.

5.2.2. Other Approaches

In the following, we present alternative approaches for the integration of compound processing and inflection prediction in SMT. The first one restricts compound processing to word alignment and can be viewed as a variant of the pre-/postprocessing approach. The other two approaches, namely *using lattice* and *synthesizing phrase tables* go beyond the modification of the training data and somewhat interfere with the SMT model.

Restriction to Word Alignment Popović et al. (2006) showed that compound splitting is beneficial for end-to-end SMT performance, even if it is only applied for word alignment. First, compounds are split prior to word alignment. Then, word alignment is performed as usual, on the original English data and the split German version of the data. The approach thus benefits from the advantages of compound splitting for word alignment, i.e. more 1:1 alignments at word level and higher frequency counts for simple words. Before the phrase extraction takes place, the positions of the English words pointing to component words of a compound are adjusted to the position of the compounds. The phrase table is then built based on this adjusted word alignment and the original English and German data. The data preprocessing for this approach is restricted to the training data. If German is the source language of the translation pair, the tuning and testing input data can remain in their original format. For the opposite translation direction, from English into German, no post-processing (e.g. in the form of compound merging) is required. This is promising for situations where a compound splitter is available, but no device for merging the compounds afterwards. In our work, we performed some initial experiments using the approach of Popović et al. (2006). However, in line with Popović et al. (2006), we soon found that the results could not improve over the more sophisticated compound splitting and merging pipelines we implemented.

Lattice-based Approach Many German compounds are ambiguous: depending on the application and the context in which the compound has occurred, multiple different splitting options might be suitable. For example, this concerns n-ary compounds with n>2, or parasite words like *"Gastraum"*, which, depending on their context can be split into either *"Gast|raum"* (= "guest|room") or *"Gas|Traum"* (= "gas|dream"). In some cases, the compound should be left unsplit, i.e. because it is lexicalised, it has a non-compositional semantics, or it coincides with a proper noun.[17]

Dyer et al. (2008) present a translation approach based on lattices, in which different splitting options are stored compactly. Their approach allows to keep multiple different splitting options of compounds during training. At testing time, the final Smt model can select the most suitable splitting for the current context. While the initial experiments of Dyer et al. (2008) focused on morpheme segmentations of Russian and Chinese, Dyer (2009) extends it to German compounds.

Phrase Table Synthesis More recently, Chahuneau et al. (2013) presented an approach to integrate inflection and derivation into Smt through phrase table synthesis. The approach is conceptually language-independent; Chahuneau et al. (2013) report on experiments for the translation pairs English to Russian, Hebrew and Swahili.

The approach works as follows: first, two translation models are trained, one on the original text and one on a stemmed version of the text. For the phrases of the latter model, inflections are predicted based on the context of the phrase in the source sentence and generated either with a rule-based analyser or an unsupervised approach. The resulting phrases are called synthetic phrases. This procedure allows to generate phrases that have not occurred in the parallel training data. The final Smt model then combines the phrase table extracted from the original text and the synthetic phrases.

[17] We introduced all of these and other characteristics of German compounds in Section 2.1.

5.3. Evaluation

In this thesis, we investigate whether knowledge from a rule-based morphological analyser can be used to improve compound processing and in turn improve SMT performance. We measure the performance of end-to-end SMT systems using standard automatic evaluation metrics. The beneficial effect of more 1:1 alignments as we motivated with intuitive examples in Figures 5.1 and 5.2 above, is not explicitly evaluated. However, larger gains in SMT performance are usually a consequence of improved phrase extractions, which in turn are based on improved word alignments.

However, there is also a benefit into the other direction: showing that this compound processing helps SMT performance can be viewed as an extrinsic evaluation of compound processing procedure. We agree here with (Demberg, 2007, , p.926), who finds that *"morphological segmentation is not of value in itself – the question is whether it can help improve results on an application"*. Alternatively, the accurracy of a compound processing approach can be evaluated with respect to a hand-crafted gold standard of correct compound splittings or, in the case of compound mergings with respect to a list of compounds. Later in this thesis, we will report the impact of compound processing in both, end-to-end SMT results and accurracies calculated on hand-crafted gold standards.

5.4. Chapter Summary

In this chapter, we motivated compound processing for SMT. We gave some intuitive examples of how compound processing helps to improve word alignment and thus SMT performance. Moreover, we described the pre-/post-processing approach we use to integrate compound processing into a classical SMT pipeline. Finally, we briefly discussed different possibilities for the evaluation of compound processing. The background part of this thesis is now complete. In the next parts, we will explore in more detail how to perform compound splitting (Part II) and compound merging (Part III), together with results from clean data and end-to-end SMT experiments.

Part II.

Compound Splitting

Motivation The focus of the second part of this thesis is on compound splitting. The aim of compound splitting is to automatically identify compounds and their component parts. As compounding is a highly productive process in German, the splitting of compounds has proven to be useful for many NLP applications that suffer from unknown words. For example, recall from Section 5.1.2, that SMT systems cannot translate compounds that have not (or only few times) occurred in the training data. Splitting such compounds into simple words which did occur in the training data makes them translatable.

Contributions In this second part of the thesis, we examine the impact of using linguistic knowledge for compound splitting as opposed to frequency-based methods of a less linguistic background. For our compound splitting approach, we make use of a rule-based morphological analyser to find linguistically motivated splitting options. These are disambiguated using corpus frequencies of words and word parts. We evaluate the performance of our approach in comparison to re-implementations of two linguistically less informed compound splitting approaches that have been widely used in SMT. One of them is solely based on corpus frequencies, the other one makes use of POS information. To do so, we use three different gold standards, two of which we created ourselves. We show that our compound splitting approach outperforms the previous approaches in all of the three gold standard evaluations. Moreover, we integrate compound splitting into an state-of-the-art SMT system for German to English, where our compound splitting, again compared to the two previous approaches, leads to improved translation performance and fewer out-of-vocabulary words.

Structure The remainder of this part is structured as follows: in Chapter 6, we describe re-implementations of two widespread compound splitting approaches for German. Our new approach, in which we consider the morphological structure of compounds prior to splitting them, is presented in Chapter 7. In Chapter 8, we give details on three different gold standard evaluations in which we compare the performance of the two previous approaches with our new approach. Finally, we integrate compound splitting into an end-to-end German to English SMT task in Chapter 9. A broader overview of other related work is given in Chapter 10.

6. Previous Approaches

In the previous chapter, we motivated the usefulness of compound splitting for statistical machine translation (SMT) in detail. To recapitulate, compounds are often missing from the training data and thus not translatable. Splitting compounds into component words that are more likely to have occurred in the training data eventually makes them translatable part by part. In this chapter, we will go into more detail on how compound splitting can be performed. We present two previous approaches to compound splitting that have recurrently been used in the past, a frequency-based approach (Koehn and Knight, 2003) and a POS-based approach (Stymne, 2008). Using re-implementations of these approaches makes them directly comparable to our morphologically-aware compound splitting approach, which we introduce in Chapter 7.

Structure The remainder of this chapter is structured as follows: we first describe the underlying concept of the two approaches, describe the disambiguation of multiple splitting options and mention tunable parameters in Section 6.1. We then present the two approaches in more detail. First, we describe the frequency-based approach of Koehn and Knight (2003) and our implementation thereof in Section 6.2. Second, the POS-based approach, for which we follow the work of Stymne (2008). However, we use a re-implementation thereof from Weller and Heid (2012), which was kindly made available to us. It is described in Section 6.3. The chapter is summarised in Section 6.4.

6.1. Common Ground

This section sets some common ground of the two compound splitting approaches which will be introduced in more detail in the subsequent sections. We motivate the common idea behind the approaches, together with some examples, we describe the disambiguation routine in case of multiple possible splitting options and finally mention parameters to create variations of the approaches.

6.1.1. Splitting Concept

In both approaches to compound splitting that we will consider in this chapter, component words are identified through substring matching: each substring of a compound that is found in a corpus is considered a valid component word.

To give an example, the component words of *"Apfelbaum"* (= "apple tree"), namely *"Apfel"* (= "apple") and *"Baum"* (= "tree") are exact substrings of the whole compound and they are very likely to be found in a corpus. Even in many cases where the modifier is **not** identical to the lemma form, the substrings will probably be found in a corpus: in the case of *"Zitronenbaum"* (= "lemon tree"), the substrings *"Zitronen"* (= "lemons") and *"Baum"* (= "tree") will most probably have occurred in the corpus, as *"Zitronen"* coincides with the plural form of the compound modifier *"Zitrone"* (= "lemon"). Recall from Section 2.2 that German compound modifier forms often coincide with either the plural or the genitive singular form of the simplex word. While the plural form tends to occur frequently enough, this need not always be the case for singular genitive forms. For such cases, the approaches usually allow for a restricted set of character modifications.

However, there are also cases where the modifier never occurs as a separate word. Consider e.g. *"Ableitungsbaum"* (= "derivation tree"), where the substring *"Ableitungs"* (= "derivation"+s) does not coincide with any form of the lemma *"Ableitung"* (= "derivation"). In the present case, a filler letter "s" must be subtracted in order to find the valid modifier *"Ableitung"* (= "derivation").

Differences The two approaches differ with respect to the restrictions they impose on substrings to be valid. The frequency-based approach only allows for a limited number of character modifications around splitting points. In contrast, the POS-based approach allows for more such modifications on the one hand, but it imposes additional restrictions regarding POS of the substrings on the other hand. To give an example, consider *"Holunderbaum"* (= "elder tree") which is split by the frequency-based approach into its most frequent substrings: *"hol+und+er+Baum"* (= "get+and+he+tree"). Having a closer look at this splitting reveals that it is not valid from a linguistic point of view. The conjunction *"und"* (= "and") and the personal pronoun *"er"* (= "he") cannot take part in German compound formation. The POS-based approach blocks splitting into such (and other) POS and thus generates a splitting into the less frequent components *"Holunder"* (= "elder") and *"Baum"* (= "tree"), which is correct in this case.

6.1.2. Disambiguation

While the two compound splitting approaches differ in the way possible splitting options are acquired (details follow below), the disambiguation of multiple splitting options is identical. In most examples given so far, the splitting into substrings was unique, but this need not always be the case. Whenever more than one possible splitting is found, the frequencies of the substrings are considered for disambiguation. We take up the *"Holunderbaum"* (= "elder tree") example from above, for which (at least) two different splitting options can be identified: *"hol|und|er|Baum"* (= "get|and|he|tree") and *"Holunder|Baum"* (= "elder|tree"). The frequencies of the substrings are the following: *"hol"* (60), + *"und"* (2,792,759) + *"er"* (489,182) + *"Baum"* (1,571) vs. *"Holunder"* (26) + *"Baum"* (1,571). Based on a formula from Koehn and Knight (2003) (p.189), geometric mean scores of the natural log frequencies of compound parts are calculated as:

$$\text{argmax}_S\Big(\sum_{p_i \in S} \frac{log(count(p_i))}{n}\Big),$$

where S = split, p_i = part, n = number of parts. The compound as a whole is also considered, it has 1 part and a minimal count of 1. We use the monolingual training data of the WMT shared task 2009[18] to derive compound and compound part frequencies. In general, the splitting that maximises the geometric mean score is selected. If the whole compound scores highest, it is left unsplit. The calculation for the two example splittings is as following:

$$\text{hol|und|er|Baum} = \frac{log(60)+log(2{,}792{,}759)+log(489{,}182)+log(1{,}571)}{4} = 4.27$$

$$\text{Holunder|Baum} = \frac{log(26)+log(1{,}571)}{2} = 3.19$$

$$\text{Holunderbaum} = \frac{1}{1} = 1$$

In this example, the erroneous splitting would be picked, due to its higher substring frequencies and the resulting higher geometric mean score.

[18] Workshop on statistical machine translation, about 227 million words, available from `http://www.statmt.org/wmt09`

6.1.3. Parameters

In the following, we present a set of parameters that allow to customize the compound splitting approaches, e.g. dependent on a new application area or language.

Scoring Metric The geometric mean score presented in the previous paragraph. This score could be substituted by another mean score (e.g. harmonic mean, arithmetic mean, see also Stymne (2008)).

Part Size The *"Holunderbaum"* (= "elder tree") example above has shown that splitting into short function words is often favoured due to their high frequency of occurrence. In order to prevent such erroneous splittings, the minimal part size (in terms of characters) could be increased. This would exclude many function words (e.g. articles, pronouns, prepositions, conjunctions, etc.).

Part Frequency It is reasonable to prevent splittings into words that have occurred only once in a corpus (possibly typos). The usage of a minimal part frequency constraint accounts for this.

Character Modifications In Section 6.1.1 above, we have already motivated the usage of character modifications for the identification of substrings. The set of possible letters to be filled in, deleted or substituted is dependent on the language the splitting is intended to be applied to. However, these modifications are easy to derive from corpus data (as done by e.g. Macherey et al. (2011)), it is not necessary to hire a linguist to compile a set of allowed character modifications for a particular language.

Stop List Depending on how the other parameters are currently set, it might be reasonable to use a stop list of frequent German words or word parts that usually do not occur in sound German compounds. Examples typically include articles, pronouns and prepositions. Such stop lists are again language-dependent, but can easily be assembled e.g. by screening through (erroneous) splitting results.

POS This parameter makes the main difference between the frequency-based and the POS-based approach. It defines the set of admitted POS classes for compounds and component words.

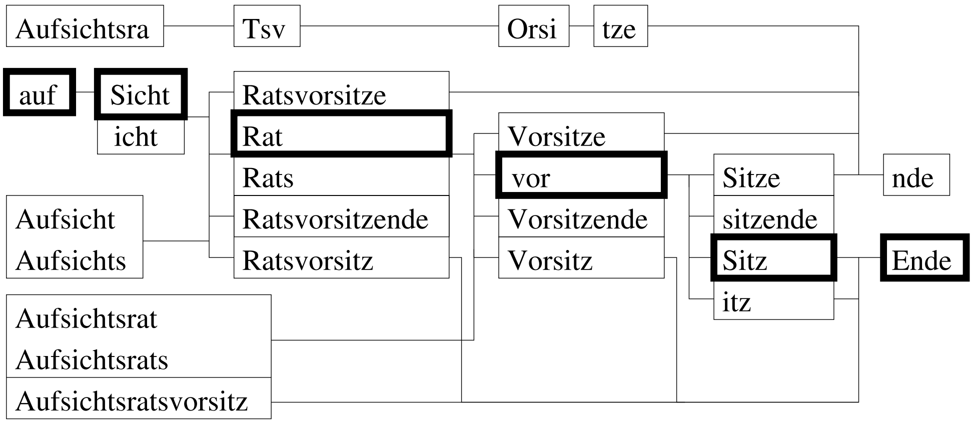


Figure 6.1.: Frequency-based splitting options for *"Aufsichtsratsvorsitzende"* (= "chairman of the supervisory board"). The correct splitting *"Aufsicht|Rat|Vorsitzende"* ("supervisory|board|chairman") is among the options, but an erronous splitting scores highest (see highlighted):*"auf|Sicht|Rat|vor|Sitz|Ende"* (= "on|sight|council|ahead|seat|end").

6.2. Frequency-based Splitting

A *frequency-based* compound splitting approach requires no linguistically motivated analysis of the compound word. Instead, substrings of compounds are identified using corpus frequencies and pre-defined lists of character modifications like the insertion of filler letters and the deletion letters. Here, we present two variants of frequency-based splitting approaches: a close re-implementation of the original approach as described by Koehn and Knight (2003) in Section 6.2.1 and an extended implementation in Section 6.2.2, with more variance on the one hand (e.g. filler letters) and more restrictions (frequency constraints, stop list) on the other hand. We already described the extended frequency-based splitting in Fritzinger and Fraser (2010). There, we use it as an contrastive baseline and compare it with our morphologically motivated compound splitting.

6.2.1. Basic

In this basic variant, compounds can be split into an arbitrary number of word parts. Every substring of the compound consisting of at least 3 characters, which occurs as a separate token in a corpus is considered a valid compound part. This allows to split *"Apfelbaum"* into the component words *"Apfel"* (= "apple") and *"Baum"* (= "tree"). Morevover, two types of filler letters are allowed: *"s"* and *"es"*. This enables the correct splitting of the compound *"Ableitung**s**baum"* into *"Ableitung"* (= "derivation") and *"Baum"* (= "tree"). Despite this limited number of filler letters, many other compounds can be correctly split, as the modifier form of many simplex words is identical to their plural or genitive form and thus also occur as substrings in the corpus from which frequencies are derived. For example, this is the case for *"Zitronenbaum"* (= "lemon trees") → *"Zitronen|Baum"* (= "lemon**s**|tree"), where the modifier coincides with the plural form of the simplex word *"Zitrone"* (= "lemon") are identical: *"Zitronen"* (= "lemons").

In Figure 6.1, we give a detailed example with all possible splitting options identified by the basic frequency-based splitting approach for *"Aufsichtsratsvorsitzende"* (= "chairman of the supervisory board"). As can be seen, the desired splitting *"Aufsicht|Rat|Vorsitzende"* (= "supervision|council|chairman") is among the options, but another splitting, namely *"auf|Sicht|Rat|vor|Sitz|Ende"* (= "on|sight|council|ahead|seat|end") maximises the geometric mean score. This happens due to high-frequency words like *"auf"* and *"vor"*, belonging to the closed word class of prepositions. From a linguistic perspective, prepositions cannot be valid compound word parts.[19] The POS-based approach we discuss in Section 6.3, filters out such word parts and thus leads to a reduced number of (erroneous) splits.

Besides over-splitting (as shown in Figure 6.1), there are also cases where compounds cannot be split, due to the restricted list of allowed morphological operations: for example, the splitting of compounds where letters of the component words were deleted to form a compound is less straightforward. The splitting might be successful in cases where the word part is identical to a (potentially different) simple word, e.g. for *"Backblechs"* (= "baking tray") → *"back|Blechs"* (= "bake$_{V-imperative}$|tray"), but often, the compounds remain unsplit e.g. *"Hüftbeugemuskulatur"* (= "hip flexor musculature"), when the deleted letter belongs to the stem of the word. In contrast to *"Hüft**e**"*, the word part *"Hüft"* is not a valid German word and thus not found in the corpus. With its limited amount of filler and deletable letters, the basic frequency-based splitting cannot access the correct splitting of the word into *"Hüft**e**|beuge**n**|Muskulatur"* (= "hip|to flex|musculature"). We address this limitation in the extended variant of the frequency-based splitting approach (allowing for more filler letters) in the next Section 6.2.2.

[19] Except their homonymous particle forms which are often identical to prepositions. See also Section 7.1.5 above.

Possible splittings					**score**
Hüft**e** (374)	beuge (35)	Mus (15)	Kula (8)	Tur (28)	3.51
hip	flex	mush	Kula	*Tur	
Hüft**e** (374)	beuge (35)	Musk (17)	Ula (5)	Tur (28)	3.45
hip	flex	*Musk	Ula	*Tur	
Hüft**e** (374)	beuge (35)	Musk (17)	Ulan (41)	Tur (28)	3.87
hip	flex	*Musk	Ulan	*Tur	
Hüft**e** (374)	beug**e** (35)	Emus (7)	Kula (8)	Tur (28)	3.36
hip	to flex	emus	Kula	*Tur	
Hüft**e** (374)	beuge**n** (536)	Mus (15)	Kula (8)	Tur (28)	4.06
hip	to flex	mush	Kula	*Tur	
Hüft**e** (374)	beuge**n** (536)	Musk (17)	Ula (5)	Tur (28)	3.99
hip	to flex	*Musk	Ula	*Tur	
Hüft**e** (374)	beuge**n** (536)	Musk (17)	Ulan (41)	Tur (28)	4.41
hip	to flex	*Musk	Ulan	*Tur	
Hüft**e** (374)	beuge (35)	Mus**e** (114)	Kula (8)	Tur (28)	3.92
hip	flex	muse	Kula	*Tur	
Hüft**e** (374)	beuge**n** (536)	Mus**e** (114)	Kula (8)	Tur (28)	4.47
hip	to flex	muse	Kula	*Tur	
Hüft**e** (374)	beuge (35)	Muskulatur (140)			4.80
hip	flex	musculature			
Hüfte (374)	**beugen (536)**	**Muskulatur (140)**			**5.71**
hip	to flex	musculature			
Hüftbeugemuskulatur					0.69

Table 6.1.: Splitting options of the extended frequency-based approach for *"Hüftbeugemuskulatur"* (= "hip flexor musculature") including word frequencies (in "()") and geometric mean scores (cf. column *score*). Deletable letters are highlighted **bold**-faced and underlined. Meaningless syllables are marked with "*" in the English gloss rows. The highest scoring splitting option is also **bold**-faced.

6.2.2. Extended

In this section, we describe a more sophisticated variant of the basic frequency-based splitting we introduced in the previous Section 6.2.1. We investigate different minimal part sizes (3-6 characters) and in order to enhance splitting recall, the extended variant covers more filler letters and also allows for some deletable letters, namely:[20]

filler letters:	*en, er, es, ien, n, nen, s*
deletable letters:	*e, n*

[20] These choices are similar to those reported in Stymne (2008) and Durgar El-Kahlout and Yvon (2010)

In addition to the disambiguation strategy introduced in Section 6.1.2, the extended approach allows filler letters to be dropped only when the part is more frequent without the letter than with it; the same holds for deletable letters to be added and hyphens. While using more filler letters allows us to split words that previously could not be split, it also leads to more erroneous splits. In order to reduce wrong splittings, low-frequent words (frequency < 5) are removed from the training corpus[21] and a stop list is used. It contains the following units, which occur in the corpus as separate words (e.g., as names, function words, etc.), and frequently occur in incorrect splittings:

stop list: *adr, and, bes, che, chen, den, der, des, eng, ein, fue, ige, igen, iger, kund, sen, ses, tel, ten, trips, ung, ver*

Table 6.1 contains all splitting options of the extended frequency-based splitting for *"Hüftbeugemuskulatur"* (= "hip flexor musculature"), together with their geometric mean scores (cf. Section 6.1.2 for details). While the correct splitting into *"Hüft**e**|beuge**n**|Muskulatur"* is found, we also observe some erroneous splittings into meaningless syllables (*"Musk"*, *"Tur"*). Recall that this compound was left unsplit by the basic frequency-based approach, due to its inability to allow for deletable letters.

6.3. POS-based Splitting

The underlying concept of what we call *POS-based* approaches to compound splitting is to take the parts of speech of words and word parts into consideration during the splitting process. This blocks splitting into closed class items (like e.g. prepositions and similar): as these are short and high-frequent words, they are compound parts favoured by the frequency-based approach. However, from a linguistic point of view, prepositions cannot take part in German compounding. The POS-based approaches identify valid word parts using part-of-speech information on whole compounds and compound part candidates. The general approach consists of two parts: i) identification of splitting options using the frequency-based approach as described in the previous section, and ii) filtering the splitting options using POS-based constraints.

Previous Work The work of Koehn and Knight (2003) is widely known for the frequency-based approach they introduce. However, this work also mentions a POS-based splitting

[21] This procedure is identical to the minimal part frequency requirement of Holz and Biemann (2008).

approach. Despite a higher splitting accuracy on held-out data, their POS-based approach was outperformed by the frequency-based approach in end-to-end SMT. Stymne (2008) extensively investigated different variants and markups and found improved performance when using the POS-based approach. In the following, we present details of a more recent implementation by Weller and Heid (2012), which was kindly made available to us. We use this implementation to approximate the performance of POS-based approaches in comparison to our splitting approach (cf. Chapter 7) and to our re-implementations of the frequency-based splitting approaches (as described in Section 6.2).

6.3.1. Data Preparation

The approach of Weller and Heid (2012) works as follows: prior to the splitting procedure, all data – namely the data set to be split and the training corpus[22] from which word and word part frequencies are derived – is processed with TreeTagger (Schmid, 1994) in order to get POS tags and lemmas. In the following, only nouns, adjectives and verbs are kept in the training corpus, as only these are considered valid compound parts. All words with other POS and all unknown words are removed from the training data. The data set to be split is further reduced to contain exclusively nouns.

Unfortunately, the training corpus still contains undesirable word units that remain after POS filtering due to tagging errors. In order to prevent splitting into meaningless word units, Weller and Heid (2012) thus perform two more filterings on the training data: first, they make use an online lexicon[23] to filter out meaningless word units of 5 characters or fewer. Then, they use the following list to remove undesired word units (similar to the extended frequency-based approach cf. Section 6.2.2):

stop list: *ale, ante, aus, barkeit, ei, eine, einen, ente, ge, gen, igkeit, ischem, ischen, ischer, lichem, lichen, licher, pro, qua, schaft, set*

After cleaning of the training corpus, frequency counts were collected. In the case of verbs these were divided by 10, as verbs only rarely constitute true compound parts but often lead to erroneous splits.

[22] Note that *training corpus* does not denote the parallel training corpus for SMT training, but refers to a big monolingual training corpus instead.

[23] `http://www.dict.cc/`

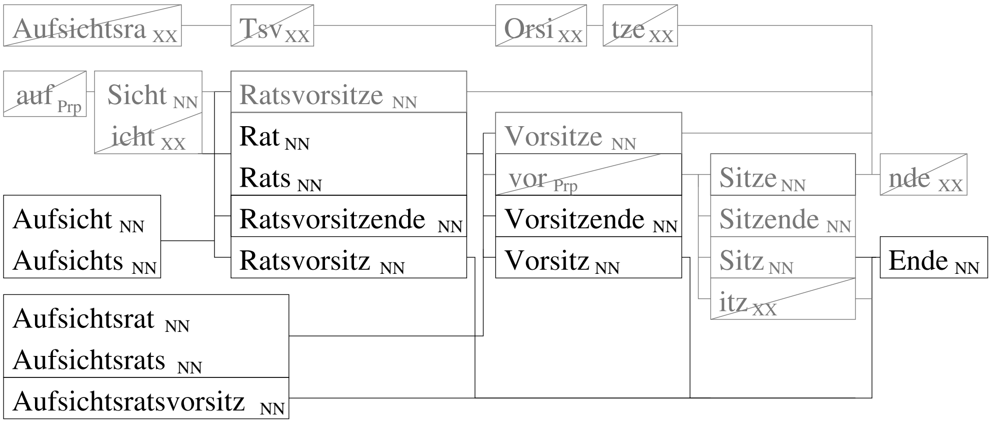


Figure 6.2.: Frequency-based splitting options for *"Aufsichtsratsvorsitzende"* (= "chairman of the supervisory board"), repeated from Figure 6.1 above and enriched with POS. Word parts that do not belong to valid POS classes are crossed out, dead paths and thus unreachable word parts are grayed.

6.3.2. Splitting Procedure

The data has now been filtered and properly prepared. Therefore, no more constraints on minimal word part size or part frequencies are imposed. However, compounds may only be split into maximal 4 word parts. Weller and Heid (2012) allow for the following filler and deletable letters:

noun filler letters:	*en, er, es, n, s*
noun deletable letters:	*e*
verb deletable letters:	*en, n*

After determining all possible splitting options, the POS of the head word (rightmost word part) is used for disambiguation: split only if the POS of the head word matches the POS of the whole compound. See Figure 6.2 for the splitting options of *"Aufsichtsratsvorsitzende"* (= "chairman of the supervisory board") remaining after application of the POS-constraints. In this case, all extracted splitting options (cf. Figure 6.2: not crossed and not grayed) meet the POS head constraint. But that need not always be the case: in Figure 6.3, we give all splitting options for *"Alternativmaßnahmen"* (= "alternative measures") before the POS head constraint has been applied. As can be seen, the option where the latter word part *"Maßnahmen"* (= "measures") is split into *"Maß|nahmen"* (= "degree|took") is blocked by the POS-based splitting approach, due to the fact that the head word *"nahmen"* is a verb and the whole compound is a noun.

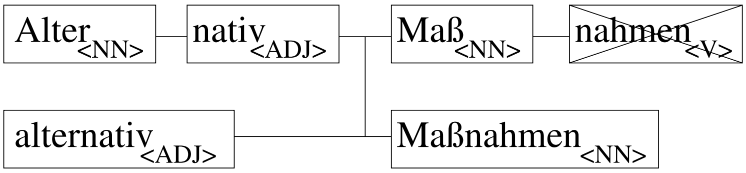


Figure 6.3.: Splitting options of *"Alternativmaßnahmen$_{<NN>}$"* (= "alternative measures") with POS of the words given in subscript. The POS approach blocks splitting into a head word whose POS does not match the POS of the whole compound (double crossed out).

6.4. Chapter Summary

In this chapter, we presented two compound splitting approaches that have recurrently appeared in recent SMT literature. We gave details of the re-implementations we used, along with illustrative examples. The portability of the two approaches to other languages requires only lingustic resources that are available for most languages (e.g. POS taggers) or can easily be compiled from scratch (e.g. a list of most common filler letters).

However, the lack of linguistic knowledge of the inherent word structure is also the main drawback of the less informed approaches we presented here, as they often lead to over-splitting of compounds but also of words which are not compounds. In the next chapter, we introduce our morphologically-aware compound splitting approach.

7. Morphological Compound Splitting

In the previous chapter, we introduced two compound splitting approaches that have been widely used in SMT in the past. Both of them rely on a minimal amount of linguistic knowledge which is restricted to a set of character modifications, a stop list, and the POS of words. Recall that one of the research questions we aim to answer in this thesis is whether linguistic knowledge can improve compound processing in SMT.

In the present chapter, we present our morphologically motivated compound splitting in detail. We make use of a rule-based morphological analyser (SMOR, Schmid et al., 2004) to find only splitting options into linguistically sound words and word parts. Moreover, compound modifiers are automatically reduced to lemmas. In case of multiple possible splitting options, we use linguistic constraints and corpus frequencies of words, word parts and combinations thereof for disambiguation. In the subsequent Chapters 8 and 9, we will show that this morphological compound splitting outperforms the less linguistically informed approaches we presented in the previous chapter, on both intrinsic and extrinsic evaluation tasks.

Previous Work In the past, there have been several previous approaches to compound splitting which made use of a rule-based morphological analyser (e.g. GerTWOL, Koskenniemi and Haapalainen (1996)). For example Nießen and Ney (2004) and Popović et al. (2006) use syntactic context for the disambiguation of multiple analyses, and Hardmeier et al. (2010) use POS-based heuristic disambiguation rules. In Fritzinger and Fraser (2010), we already described the compound splitting approach based on SMOR, which we will present in this chapter. It disambiguates SMOR analyses using corpus frequencies. In contrast to Fritzinger and Fraser (2010), the approach now works on token level (instead of type level). This allows to decide context-based whether a proper name that coincides with a German common noun is to be split or not.

```
(1) no modification required:

> Hausstaub
Haus<NN>Staub<+NN><Masc><Nom><Sg>
Haus<NN>Staub<+NN><Masc><Acc><Sg>
Haus<NN>Staub<+NN><Masc><Dat><Sg>

(2) insertion of a filler letter:

> Kartenhäuser
Karte<NN>Haus<+NN><Neut><Nom><Pl>
Karte<NN>Haus<+NN><Neut><Gen><Pl>
Karte<NN>Haus<+NN><Neut><Acc><Pl>
```

```
(3) deletion of letters:

> Backblech
backen<V>Blech<+NN><Neut><Nom><Sg>
backen<V>Blech<+NN><Neut><Acc><Sg>
backen<V>Blech<+NN><Neut><Dat><Sg>

(4) transformation of letters:

> Kriterienliste
Kriterium<NN>Liste<+NN><Fem><Acc><Sg>
Kriterium<NN>Liste<+NN><Fem><Gen><Sg>
Kriterium<NN>Liste<+NN><Fem><Nom><Sg>
Kriterium<NN>Liste<+NN><Fem><Dat><Sg>
```

Figure 7.1.: Examples of Smor analyses for morphological operations that are required for combining simple words into compounds. (1) *"Hausstaub"* (= "house dust"), (2) *"Kartenhäuser"* (= "card houses"), (3) *"Backblech"* (= "baking tray"), (4) *"Kriterienliste"* (= "criteria list"), with NN = noun, V = verb, Masc = masculine, Fem = feminine, Neut = neuter, Acc = accusative, Gen = genitive, Nom = nominative, Dat = Dative, Sg = singular, Pl = plural

Structure The remainder of this chapter is structured as follows: In Section 7.1, we give technical details on how to use a rule-based morphological analyser (Smor, Schmid et al. (2004))) to derive possible splitting options. The disambiguation procedure which applies in case of multiple splitting options is described in Section 7.2.

7.1. Linguistic Analysis

In this section, we give some technical compound-related details of how to use Smor to derive possible splitting options. We first illustrate the identification of word boundaries in Section 7.1.1. In Section 7.1.2 we give details on how to proceed with bound word parts like e.g. derivational affixes that cannot freely occur in fluent German. Thereafter, we describe how Smor can be used for lemmatisation of modifiers or heads in Section 7.1.3 and finally we briefly discuss the importance of true-casing text prior to Smor analysis in Section 7.1.4.

7.1.1. Identification of Word Boundaries

We already introduced basic functionalities in Section 3.2 above. Recall that SMOR covers the inflection, derivation, and composition of German word formation. Each input word is reduced to its lemma and, in the case of compounds, input words are reduced to their component words and word parts. SMOR relies on a huge lexicon, which is enriched with information on the morphological operations that are necessary in order to form a sound German compound from two (or more) simple component words. In contrast to the previous approaches which we described in the Chapter 6, there is thus no need to define lists of possible filler or deletion letters nor to specify rules for character transformations. All this knowledge is encoded in SMOR's lexicon. Some examples of different morphological operations that are required to put two simple words together are given in Figure 7.1. SMOR's analysis format for nouns is to be read as follows:

```
lemma1<POS>lemma2<POS><Gender><Case><Number>
```

Figure 7.1 shows that SMOR returns several analyses for each of the input words. It can be seen that these are structurally identical and differ only in the case feature. These are typical examples for German case syncretism. However, as differences in the *case* of the (head) word are irrelevant for compound splitting, we neglect all analyses which are structurally identical in our further processing. We will not repeat them in the examples we give in the remainder of this thesis.

The analyses given in Figure 7.1 clearly reveal the component words of each compound in that these are separated from the rest of the analysis with an own POS-tag. We use these POS tags within an analysis to determine split points,
e.g. *"Hausstaub"* (= "house dust"):

```
Haus<NN>Staub<+NN><Masc><Nom><Sg> --> Haus|Staub
```

7.1.2. Bound Morphemes

The examples given in Figure 7.1 show that word parts are separated by POS-tags when analysed by SMOR. This applies to component words of compounds (as above), but also to other word parts like e.g. suffixes, prefixes and particles. However, such derivational suffixes and prefixes are bound morphemes that usually cannot occur freely, but only in conjunction with a word stem. See Figure 7.2 for SMOR analyses of *"verhandelbar"* (=

```
> verhandelbar
ver<VPREF>handeln<V>bar<SUFF><+ADJ><Pos><Adv>
(1) --> ver|handeln|bar
(2) --> verhandelbar

> Bearbeitungsgebühr
be<VPREF>arbeiten<V>ung<NN><SUFF>Gebühr<+NN><Fem><Nom><Sg>
(1) --> be|arbeiten|ung|Gebühr
(2) --> Bearbeitung|Gebühr
```

Figure 7.2.: Smor example analyses containing bound morphemes, i.e. prefixes and suffixes, that should not be split. (1) *“verhandelbar”* (= “negotiable”), (2) *“Bearbeitungsgebühr”* = (“processing fee”), with VPREF = verb prefix, V = verb, SUFF = suffix, ADJ = adjective, Pos = positive (comparation form), Adv = adverbial usage, NN = noun, Fem = feminine, Nom = nominative, Sg = singular. Structurally identical analyses are omitted in order to save space.

“negotiable”) and *“Bearbeitungsgebühr”* (= “processing fee”) which both contain bound morphemes, namely the verb prefixes *“be-”* and *“ver-”* (marked *“<VPREF>″*), as well as the suffixes *“-bar”* and *“-ung”* (marked *<SUFF>*). During the splitting process, we extract all tag-separated words from the Smor analysis, but separations into bound morphemes are blocked based on their POS tags. Note however, that verb particles can be split from their verbs under certain circumstances, see Section 7.1.5 for details.

7.1.3. Lemmatisation

Depending on the actual application for which compound splitting is required, it may (or may not) be of interest to lemmatise compound modifiers and/or compound heads. As can be seen from the examples given in Figure 7.1, both of these word part types, modifiers and heads, are automatically reduced to their lemmas, when analysed with Smor, e.g. *“Kartenhäuser”* (= “card houses”) → *“Karte|Haus”* (= “card house”).

It is thus straightforward to extract lemmatised compound modifiers and heads from this analysis. For our application of compound splitting in German to English Smt, it is reasonable to reduce modifiers in order to be able to generalise over word occurrences in modifier vs. head position. In contrast, the compound heads should remain fully inflected, because through lemmatisation of head words, we lose some contextual morphological information, e.g. the number of the word, and this loss may lead to confusions in word

```
> Kartenhäuser
Karte<>:n<NN>:<>H:ha:äus<+NN>:<><Neut>:<><>:e<>:r<Nom>:<><Pl>:<>

> Backblechs
b:Backe:<>n:<><V>:<>B:blech<+NN>:<><Neut>:<><Gen>:<><Sg>:<><>:s
```

Figure 7.3.: Running SMOR with the ***"-b"*** flag reveals the internal two-level structure of its lexicon entries. We use this output format for our splitting procedure. See Figure 7.4 a more human-readable version of these two analyses.

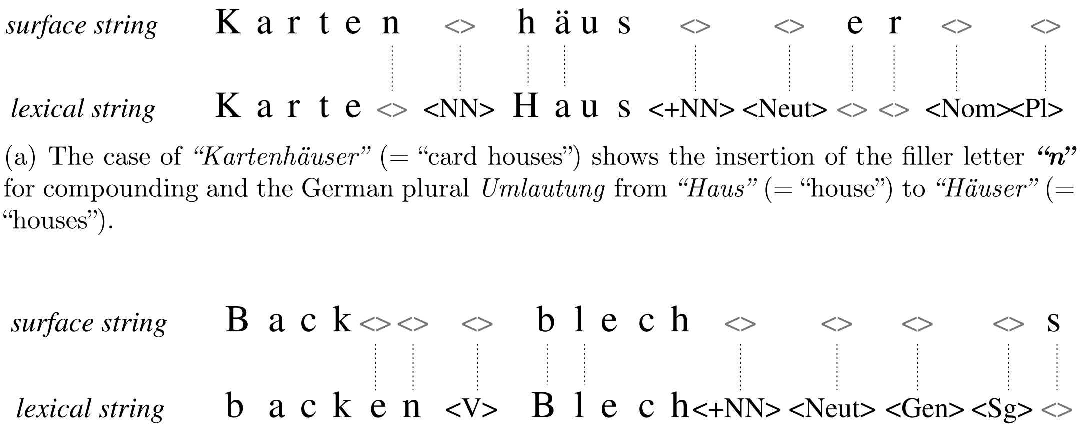
alignment (e.g. when both English word forms "house" and "houses" are both aligned to German *"Haus"*).[24] For the above example *"Kartenhäuser"*, the desired split analysis for our application is thus *"Karte|Häuser"* (= "card houses").

SMOR's lexicon entries encode both of the morphological analysis levels:[25] the lexical level (lemmatised) and the surface level (fully inflected). Running SMOR with the ***"-b"*** flag reveals this internal two-level structure of each analysis. Two examples of such analyses are given in Figure 7.3. Having access to this internal structure facilitates the extraction of compound modifiers in either their reduced or fully specified format. If desired, the procedure could easily be adapted to extract both word part types on lexical or surface level, respectively.[26] However, as the format shown in Figure 7.3 is not very human-readable, we will show examples of SMOR analyses in their *default* format (as shown in Figure 7.1) throughout the remainder of this thesis. A visualisation of the two-level analyses of Figure 7.3 is given in Figure 7.4.

[24] Note that for the opposite translation direction, English to German, where we combine compound processing with inflection prediction, we will also reduce head words to their lemma form, but keep the *number* information in the feature set assigned to each lemma. See Section 11 for details.

[25] See Section 3.2 above for details on two-level morphology.

[26] *"Kartenhäuser"* can be split into *"Karte|Haus"* (both lemmatised), *"Karte|Häuser"* (modifier lemmatised, head not), *"Karten|Haus"* (modifier in surface form, head lemmatised), *"Karten|Häuser"* (both in surface form).


surface string
K a r t e n <> h ä u s <> <> e r <> <>
lexical string
K a r t e <> <NN> H a u s <+NN> <Neut> <> <> <Nom><Pl>
surface string
B a c k <> <> <> b l e c h <> <> <> <> s
lexical string
b a c k e n <V> B l e c h <+NN> <Neut> <Gen> <Sg> <>

(a) The case of *"Kartenhäuser"* (= "card houses") shows the insertion of the filler letter ***"n"*** for compounding and the German plural *Umlautung* from *"Haus"* (= "house") to *"Häuser"* (= "houses").

(b) In contrast, the verbal ending ***"en"*** has to be deleted in order to form a sound compound in the case of *"Backblechs"* (= "baking tray"), and a genitive ending ***"s"*** is inserted.

Figure 7.4.: Visualisation of the two-level representations of Smor's lexicon entries for the two analyses given in Figure 7.3

7.1.4. True-casing

It is important to note that Smor works case-sensitively on word level, i.e. it returns different analyses for the same word in its upper-cased vs. lower-cased variants. In written German text, regular nouns are upper-cased, together with proper names and all sentence-initial words. Note that the latter two occur upper-cased in many other languages as well, e.g. English, French, and Swedish.

Wrong casing variants, e.g. due to sentence-initial adjectives, may either not be analysed by Smor at all, or, in case they coincide with regular nouns, they may erronously be split. Consider e.g. the adjective *"amerikanische"* (= "American"), that should not be split when occurring as a (lower-cased) adjective, but coincides in its upper-cased variant with the – semantically implausible – noun *"Amerikanische"*, which could be split into *"Amerika|Nische"* (= "America|niche"). It is thus important to true-case all words before analysing them with Smor. True-casing can either be performed by making use of the POS of a word (using a POS-tagger) or by chosing the most frequent casing variant that has occured in a corpus.

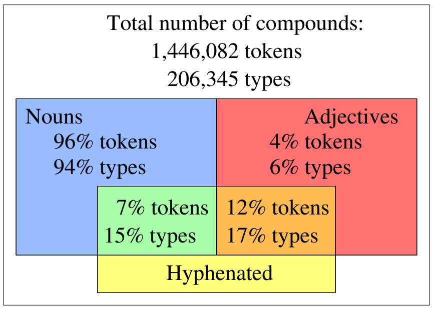


Figure 7.5.: Distribution of noun, adjective and hyphenated compounds derived from the German section of Europarl v4 (containing roughly 39 million tokens).

7.1.5. Beyond Noun Compounds

So far, we presented only examples of analysing noun compounds, a group to which most German compounds belong to. In general however, we allow all kinds of compounds to be split, including adjectives, particle verbs and hyphenated compounds In our basic approach, the only exception are proper names that can also be used as common nouns. In this section, we will briefly describe how we will deal with these kinds of compounds in our SMOR-based splitting procedure.

Adjectives Adjective compounds can be generated productively in German and we thus include them into our splitting procedure. In general, the splitting of adjectives happens straightforwardly to the splitting of nouns as described in the preceding section. Figure 7.5 illustrates the quantitative proportions of noun and adjective compounds.

In our data, we observed productive adjectives with two different kinds of compound heads: i) genuine adjectives (e.g. *"auswanderungswillig"* = "willing to emigrate") or ii) adjectives derived from verb participles (e.g. *"lichtdurchflutet"* = "flooded with light").

Particle Verbs Even though particle verbs are a closed word class (no new particle verbs can be generated from scratch), we split particles from their verbs prior to word alignment. This allows us to translate verb+particle combinations that have not been seen in the training data, if their parts have been seen independently or in combinations with other verbs and particles. We give three different examples of particle verb usage in Figure 7.6. In German, particles can optionally be split from their verbs (cf. Fig. 7.6 (A) vs. (B)), depending on the position and the tense of the verb. Note that Nießen and Ney (2000) do exactly the opposite: they attach particles that occur separated from their

(A) particle verb occurs split:	DE:	es	kühlte	schnell	ab
→ no splitting required		(*it*	*cooled*	*rapidly*	*down*)
	EN:	it	rapidly	cooled	down
(B) particle verb can be split:	DE:	weil	es	schnell	abkühlte
		(*because*	*it*	*rapidly*	*down-cooled*)
→ split particle from verb	**split**:				**ab** | **kühlte**
	EN:	because	it	rapidly	cooled down
(C) particle cannot be split:	DE:	es	ist	schnell	abgekühlt
→ no splitting		(*it*	*is*	*rapidly*	*down-cooled*)
	EN:	it	has	rapidly	cooled down

Figure 7.6.: Three different cases of particle verb usage: (A) particle occurs split, (B) particle can be split and (C) particle cannot be split.

verbs back to the verb prior to translation. In contrast, for particle verbs in **present perfect tense** the particles can never occur separated from their verbs (as in *“ab**ge**kühlt”* of Fig. 7.6 (C)): *“*es ist gekühlt schnell ab”* (= “*it has cooled rapidly down”). We thus split particles from their verbs only if they do not occur in present perfect tense. In Smor such forms are marked by the feature tag *<PPast>*.

Proper Nouns There are a number of words that can, dependent of their context, be either interpreted as proper nouns or common nouns. Examples include the family name *“Baumeister”* (= “master builder”) and the regional beer brand name *“Dinkelacker”* (= “spelt field”), for which Smor returns both the proper noun and the common noun analysis. Our Smor-based splitting allows to optionally take into account context-sensitive POS-tags in order to split such words only when used as a common noun in the present context. POS-tags can either be obtained from a POS-tagger or a syntactic parser. We usually use BitPar (Schmid, 2004). This is in line with previous work of Nießen and Ney (2000) who also used a parser to disambiguate morphological analyses of a rule-based morphology for compound splitting.

Hyphenated Compounds Moreover, both nouns and adjective compounds can occur in hyphenated format. If they do, they require some special treatment when being analysed with Smor. In Figure 7.5, we illustrate the quantitative proportion of hyphenated compounds. We give examples of hyphenated compounds together with their Smor analyses in Figure 7.7. As can be seen, at least the head word of a hyphenated com-

```
(1) analysis without hyphen                    (3) modifiers are not analysed

analyze> Hausstaubmilbe                        analyse> 1A!-Milbe
Haus<NN>Staub<NN>Milbe<+NN><Fem><Nom><Sg>     {1A!}-<TRUNC>Milbe<+NN><Fem><Nom><Sg>

(2) analysis with hyphen                       (4) no analysis for unknown heads
analyze> Hausstaub-Milbe                       analyse> Hausstaub-Millbe
{Hausstaub}-<TRUNC>Milbe<+NN><Fem><Nom><Sg> no result for Hausstaub-Millbe
```

Figure 7.7.: *"Hausstaubmilbe"* = "house dust mite"; SMOR returns analyses only if the head word is included in the lexicon (1-3), compound modifiers in hyphenated words are left unanalysed, irrespective of their occurrence in the lexicon as in (2) or nonsense (3). *"Millbe"* in (4) is a typos of *"Milbe"* (= "mite").

pound must be included in the lexicon. Otherwise the compound is cannot by analysed by SMOR. Whenever this condition is fulfilled, we can extract possible split points from the SMOR analyses as usual. Note however that hyphenated modifiers are always left unanalysed by SMOR and thus remain unsplit even if they are compounds themselves (cf. the modifier *"Hausstaub"* = "house dust" in example (2) of Figure 7.7).

While hyphenated compounds with unknown head words are left completely *unanalysed* by SMOR (as it may happen due to typos, e.g. *"Millbe"* instead of *"Milbe"* = "mite", cf. Figure 7.7 (4)), we implemented a fallback strategy that splits hyphenated compounds at their hyphen, even without SMOR analysis. For the case of *"Hausstaub-Millbe"*, this allows word alignment to generalise at least over the occurrences of the modifier *"Hausstaub"* (= "house dust").

Note also that many proper nouns contain hyphens and are not part of SMOR's lexicon, e.g. *"Wal-Mart"* or *"Al-Qaeda"*. In order to prevent these hyphenated words from being split, we extended the fallback splitting strategy to split unanalysed hyphenated compounds only if they occurred more than 5 times in our training corpus.

7.2. Disambiguation

In the previous sections, we already mentioned that structurally identical analyses differing only in one feature (e.g. *case*, cf. Figure 7.1 above) are discarded and that we filter out splittings into bound word parts (cf. Figure 7.2). However, the remaining analyses still have to be disambiguated in order to fine one *best* splitting option.

Previous Work Demberg (2006), who used SMOR for letter-to-phoneme conversion, reports that she found on average 2.4 different segmentations per word. In the past, the disambiguation of morphological analysers was often performed using context-sensitive POS-tags from a parser (Nießen and Ney, 2000), POS-based heuristic disambiguation rules (Hardmeier et al., 2010) or by training classifiers, see e.g. Habash and Rambow (2005) for Arabic or Yuret and Türe (2006) for Turkish.

Structure In the following, we describe our disambiguation approach, which consists of two steps: first, we restrict the analysis depth in Section 7.2.1, and finally, we use corpus-derived frequencies to disambiguate the remaining splitting options in Section 7.2.2.

7.2.1. Analysis Depth

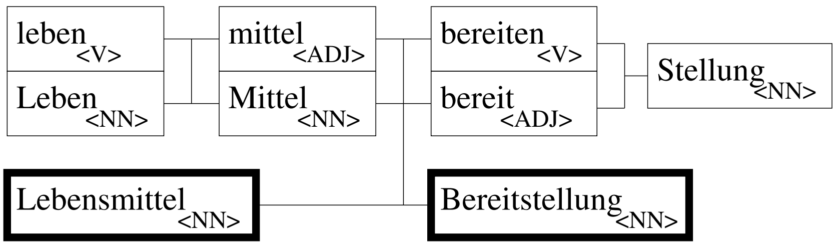
SMOR returns a deep morphological analysis for each word. However, for the present application of compound splitting in SMT, the aim is to find one best splitting option so that the each component word of the German compound ideally corresponds to one English words. A high-level linguistic analysis is thus mostly sufficient. In contrast, for applications e.g. in the field of lexical semantics, where SMOR could be used to approximate the meaning of a word or compound, the deep analysis level might be more desirable.

Filter flag Note that SMOR features an internal filter ("***-d***" for disambiguation) keeping only high-level analyses with the least number of morphemes. This leaves fully lexicalised compounds unsplit. As a consequence, opaque compounds are left unsplit, if they are covered by SMOR's lexicon.[27] An early approach by Rackow et al. (1992) pursues a similar strategy in that all words that have an own entry in a hand-crafted lexicon are left unsplit. This procedure is also in line with Schiller (2005), who found that human readers, when faced with output of an unweighted morphological analyser (similar to SMOR) often prefer splittings into the smallest number of parts. Finally, Demberg (2006) used different settings of SMOR to find optimal segmentations for the task of grapheme-to-phoneme conversion with and without the restricted analysis depth option. Consider the example *"Lebensmittelbereitstellung"* (= "food supply") in Figure 7.8, where we summarise many structurally different analyses of different depths. Using the "***-d***" flag for restricted analysis depth when analysing *"Lebensmittelbereitstellung"* (= "food supply"), only the

[27] See also Section 5.1.1 on compositionality (page 52 above).

```
> Lebensmittelbereitstellung
leben<V><NN><SUFF>Mittel<NN>Bereitstellung<+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>Mittel<NN>be<VPREF>reiten<V>Stellung<+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>Mittel<NN>be<VPREF>reiten<V>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>Mittel<NN>bereit<ADJ>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>Mittel<NN>bereit<ADJ>Stellung<+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>Mittel<NN>bereit<VPART>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>mittel<ADJ>be<VPREF>reiten<V>Stellung<+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>mittel<ADJ>be<VPREF>reiten<V>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>mittel<ADJ>bereit<ADJ>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>mittel<ADJ>bereit<ADJ>Stellung<+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>mittel<ADJ>bereit<VPART>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
leben<V><NN><SUFF>mittel<ADJ>Bereitstellung<+NN><Fem><Nom><Sg>
Leben<NN>mittel<ADJ>Bereitstellung<+NN><Fem><Nom><Sg>
Leben<NN>mittel<ADJ>be<VPREF>reiten<V>Stellung<+NN><Fem><Nom><Sg>
Leben<NN>mittel<ADJ>be<VPREF>reiten<V>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Leben<NN>mittel<ADJ>bereit<ADJ>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Leben<NN>mittel<ADJ>bereit<ADJ>Stellung<+NN><Fem><Nom><Sg>
Leben<NN>mittel<ADJ>bereit<VPART>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Leben<NN>Mittel<NN>Bereitstellung<+NN><Fem><Nom><Sg>
Leben<NN>Mittel<NN>be<VPREF>reiten<V>Stellung<+NN><Fem><Nom><Sg>
Leben<NN>Mittel<NN>be<VPREF>reiten<V>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Leben<NN>Mittel<NN>bereit<ADJ>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Leben<NN>Mittel<NN>bereit<ADJ>Stellung<+NN><Fem><Nom><Sg>
Leben<NN>Mittel<NN>bereit<VPART>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Lebensmittel<NN>Bereitstellung<+NN><Fem><Nom><Sg>
Lebensmittel<NN>be<VPREF>reiten<V>Stellung<+NN><Fem><Nom><Sg>
Lebensmittel<NN>be<VPREF>reiten<V>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Lebensmittel<NN>bereit<ADJ>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
Lebensmittel<NN>bereit<ADJ>Stellung<+NN><Fem><Nom><Sg>
Lebensmittel<NN>bereit<VPART>stellen<V>ung<SUFF><+NN><Fem><Nom><Sg>
```

(a) Summary of structurally different SMOR analyses. Note that SMOR analyses are not ranked according to analysis depth.



(b) Illustration of possible splittings based on the above analyses. Splittings into bound word parts like prefixes (*“be-”*) or suffixes (*“-ung”*) are blocked.

Figure 7.8.: Deep morphological analysis of *“Lebensmittelbereitstellung”* (= “food supply”), with *“leben”* = “to live”, *“Leben”* = “life”, *“Lebensmittel”* = “food”, *“Mittel”* = “average/means”, *“mittel”* = “mid”, *“bereiten”* = “to prepare”, *“reiten”* = “to ride”, *“bereit”* = “ready”, *“stellen”* = “to put”, *“Stellung”* = “position”, *“Bereitstellung”* = “supply”

analysis: `Lebensmittel<NN>Bereitstellung<+NN><Fem><Nom><Sg>` remains, and no more disambiguation is required. This internal filtering is helpful to prevent unwanted splittings into too many word parts, e.g. splitting *"Lebensmittelbereistellung"* into *"Leben|Mittel|bereit|Stellung"* (= "life|means|ready|position"), but the correct splitting sometimes cannot be found due to lexicalised compound parts: *"Lebensmittelpunkt"* (= "centre of life"), which should be split into *"Leben|Mittelpunkt"* (= "life|centre"), but the only analysis returned when using the internal disambiguation filter is *"Lebensmittel| Punkt"* (= "food|point"), because *"Lebensmittel"* is lexicalised in SMOR. While this splitting is morphologically sound, it is semantically implausible.

Hierarchy SMOR's implementation (as a finite-state-transducer, see Section 3.2 for details) does not allow for a hierarchically structured segmentation of compound words that consist of more than two component words. For example, the semantics of the German compound *"Turbinenpassagierflugzeug"* varies depending on its context and whether a right-branching *(Turbine(Passagier|Flugzeug))* (= (turbine(passenger|air- craft))) or a left-branching *((Turbine|Passagier)Flugzeug)* (((turbine|passenger)aircraft)) word structure is assumed. SMOR's analysis of the word only reveals that it consists of the three parts *"Turbine"* (= "turbine"), *"Passagier"* (="passenger") and *"Flugzeug"* (= "aircraft").

7.2.2. Word part Frequencies

After having deleted structurally identical analyses (only differing in features like e.g. *case* or *number*), and restricting the analysis depth, we finally use corpus frequencies to disambiguate the remainder set of analyses in order to select one splitting option. The disambiguation procedure we use is essentially the same as described in Section 6.1.2 above. We briefly repeat it here for readability. We follow Koehn and Knight (2003), who used the geometric mean of substring frequencies to find optimal split points.

We calculate the geometric mean scores of splitting option based on the natural log frequencies of word parts given by the SMOR analyses.[28] The splitting that maximises the geometric mean score is picked. The following formula is adapted from (Koehn and Knight, 2003, , p.189):

$$\text{argmax}_S(\sum_{p_i \in S} \frac{log(count(p_i))}{n})$$

[28] We use the monolingual training data of the WMT shared task 2009, to derive word and word part frequencies. It consists of about 146 million words. `http://www.statmt.org/wmt09`

Possible splittings			score
alternativ (210) *alternative*$_{ADJ}$	stromern (3) *to roam*	Zeuger (1) *creator*	2.18
alternativ (210) *alternative*$_{ADJ}$	Strom (5,499) *power*	Erzeuger (1,473) *producer*	7.11
alternativst (1) *most alternative*	Rom (5,132) *Rome*	Erzeuger (1,473) *producer*	0
Alternative (5,036) *alternative*$_{NN}$	stromern (3) *to roam*	Zeuger (1) *creator*	3.20
Alternative (5,036) *alternative*$_{NN}$	**Strom (5,499)** *power*	**Erzeuger (1,473)** *producer*	**8.14**
alternativ (210) *alternative*$_{ADJ}$	Stromerzeuger (136) *power producer*		5.17
alternativst (1) *most alternative*	Romerzeuger (1) *Rome producer*		0
Alternative (5,036) *alternative*$_{NN}$	Stromerzeuger (136) *power producer*		6.71
alternativstromern (1) *to roam alternatively*		Zeuger (1) *creator*	0
Alternativstrom (1) *alternative power*		Erzeuger (1,473) *producer*	0

Figure 7.9.: All possible splittings and recombinations for *"Alternativstromerzeuger"* (= "alternative power producer") with restricted analysis depth, including word frequencies (in "()") and geometric mean scores (cf. column *score*).

with S = split, p_i = part, n = number of parts. Whenever a word part has not occured in the data (thus having a frequency of 0), the geometric mean score for this splitting option was set to 0. In a second step, we generate all possible re-combinations of these word parts and calculate the geometric mean scores for those as well.

A detailed example is given in Figure 7.9 where SMOR, when used with the disambiguation flag, still returns five structurally different analyses for *Alternativstromerzeuger* (= "alternative power producer"). We give corpus frequencies (in brackets) and geometric mean scores (rightmost column) for all of the five splitting options and recombinations of word parts within them. It can be seen that the splitting into *"Alternative|Strom|Erzeuger"* (= "alternative|power|producer") scores highest and is thus picked. However, in cases where the natural log frequency of the word as a whole exceeds the geometric mean score of the splitting options, the word is left unsplit. This concerns e.g lexicalised combinations that lost their compositionality over time. Consider the case of *"Armaturenbrett"* (= "dashboard"), which occurred 211 times and scored 5.35, while its splitting *"Armatur"* (15), *"Brett"* (423) (= "armature|board") yields a score of only 4.37.

7.3. Chapter Summary

In this chapter, we introduced our morphological compound splitting procedure in great detail. We have explained the analysis format of Smor, along with numerous examples, described our disambiguation strategy in case of multiple analyses and motivated the usage of two featured flags: "***-b***" reveals the internal two-level structure of the analyses and enables thus the extraction of lemmatised versions vs. surface forms of the word parts and "***-d***" reduces the analysis depth which is favourable for Smt. In Chapters 8 and 9, we will compare splittings with and without this flag and show that this assumption holds. Moreover, we show that the morphologically-aware compound splitting outperforms less informed approaches in terms of splitting accuracy and translation quality.

8. Gold Standard Evaluation Results

The previous chapters dealt with different approaches to compound splitting for SMT. In this chapter we evaluate the accuracy of the different compound splitting approaches with respect to three manually annotated gold standards. The results show that our linguistically motivated compound processing, which we presented in the previous chapter, outperforms the two less informed approaches, which we introduced in Chapter 6. In the next chapter, this improvement of the lingustically motivated approach over the previous approaches will be confirmed in an end-to-end SMT system.

Annotation Details In general, the manual creation of a gold standard is time-consuming task. Due to the fact that the annotation of compounds happens on word level, it is much easier than for example manual syntactic annotations or word alignments. To annotate the component words of a compound is usually a straightforward task for a native speaker of German: starting from a clean data set, the human annotator identifies compounds and manually annotates their most plausible split points. In the course of the annotation process, the **hierarchical structure** of compounds was considered for the splitting decision (see the example of *"Untersuchungshäftling"* on page 101 below), but no branching structure was annotated for n-ary compounds of n>2.

Besides the structure, the **compositionality** of a compound is taken into account: only compositional compounds are split by the human annotator, even though many non-compositional compounds consist of two (or more) words. An example is *"Kopfsalat"* (= "lettuice", lit. "head|salad") which remains unsplit in contrast to the fully compositional *"Bohnensalat"* (= "bean|salad"). Non-compositional compounds are challenging for all of the compound splitting approaches, as none of them explicitly checks for compositionality prior to splitting. To overcome this problem, we plan to integrate semantic knowledge into the compound splitting process in the future.[29]

[29] However, opaque compounds are often lexicalised. Using the filter-flag "-d" for SMOR in the course of splitting, prevents fully lexicalised compounds from being split. See also Section 5.1.1 on p. 52 and

Previous Work Gold standards have been widely used in previous work to evaluate the accurracy of a compound splitting approach, even though, as Holz and Biemann (2008) claim, "there is no publicly available standard dataset for German compound noun decomposition". As a consequence, some groups created their own gold standards customised for their applications, e.g. (Monz and de Rijke, 2001), (Schiller, 2005), (Alfonseca et al., 2008a) and (Holz and Biemann, 2008). In contrast, Demberg (2007) used available data from CELEX for her evaluation and Macherey et al. (2011) find a gold standard evaluation not neccessary, as long as there is a measurable improvement of translation quality.

While most previous work on gold standard evaluations for compound splitting report mainly on accuracy scores, we also give details on the creation of the gold standards we use, perform detailed error analyses and give many illustrative examples. Note that we already used the two first gold standards presented in this chapter in Fritzinger and Fraser (2010). There, we evaluated a previous version of our compound splitting approach with respect to these gold standards.

Structure The remainder of this chapter is structured as follows: we first summarise the compound splitting approaches we investigate in Section 8.1 and present the evaluation metrics we will use in Section 8.1.2. Then, we present three different gold standards, which all have different characteristics. We describe how their creation and calculate accuracy scores for all of the investigated splitting approaches. The first gold standard described in Section 8.2 is token-based and includes translational correspondences, whereas the second one in Section 8.3 is type-based without context. While we created these two gold standards ourselves, we present also an external domain-specific gold standard in Section 8.4. We compare the accurracies of all gold standards in Section 8.5 and give a detailed error analysis in Section 8.6, where we present examples for typical errors of each of the approaches. Finally, we summarise our findings in Section 8.7.

the paragraph *filter flag* in Section 7.2.1, p. 86 above.

name	subword restrictions	filler letters	delet. letters	stop words	other	to be split	details in Section
basic freq.	min. 3 character	limited	no	no	no	all	6.2.1
extended freq.	min. 4 character	extend.	yes	yes	part freq. ≥ 3	all	6.2.2
POS	corpus cleaning	extend.	yes	yes	pos-constraints	nouns	6.3
Smor	Smor	n.a.	n.a.	no	deep analysis	all	7.1
Smor -d	Smor	n.a.	n.a.	no	flat analysis	all	7.2.1
Smor -d NN	Smor	n.a.	n.a.	no	flat analysis	nouns	7.2.1

Table 8.1.: Tabular overview of the compound splitting approaches we compared and evaluated with respect to different gold standards.

8.1. Experimental Settings

This section contains details concerning the parameter settings of the splitting approaches we investigated and presents the metrics we use for evaluation.

8.1.1. Experiments

An overview of the different splitting approaches we evaluated using gold standards is given in Table 8.1. We distinguish between three main types of different approaches: 1) frequency-based (basic and extended variant), 2) POS-based and 3) morphologically-aware (= our approach). For the extended frequency-based approach, we follow Durgar El-Kahlout and Yvon (2010) and set the minimal character size for compound parts to 4. For our approach, we also consider different variants: "-d" indicates that we use only the highest morphological analysis level: e.g. there is no further decomposition into the verbs of verbal nominalisations like *"Rechnung"* (= "calculation") into *"rechnen$_V$ + -ung$_{SUFF}$"* (= "calculate$_V$ + -ion$_{SUFF}$"). Moreover, with "NN" we also investigate a variant where only noun compounds are split (no adjectives, no particle verbs). More details on the approaches can be found in Chapters 6 and 7.

Note that the differences of the splitting approaches in Table 8.1 only concern the way that splitting options are obtained. The last step of disambiguating remaining options using corpus frequencies is identical for all approaches (cf. Chapter 6, page 65 for details). We use the German section of the monolingual training data, of the EACL 2009 workshop on statistical machine translation[30] (∼227 million words) to derive word and word part frequencies for all approaches.

[30] http://www.statmt.org/wmt09

Name	Compound	Gold Splitting	System Splitting	Counts
correct split	Ortszeit *local time*	Ort|Zeit *location|time*	Ort|Zeit *location|time*	128
correct not	Ausstieg *exit*	Ausstieg *exit*	Ausstieg *exit*	4,730
wrong split	Ausstieg *exit*	Ausstieg *exit*	aus|stieg *off|got*	116
wrong not	Ortszeit *local time*	Ort|Zeit *location|time*	Ortszeit *local time*	15
wrong faulty	Goldbarren *gold ingot*	Gold|Barren *gold|ingot*	Gold|Bar|Ren *gold|bar|reindeer*	11
precision	$\frac{correct\ split}{correct\ split\ +\ wrong\ faulty\ +\ wrong\ split}$		$\frac{128}{128\ +\ 11\ +\ 116}$	50.20%
recall	$\frac{correct\ split}{correct\ split\ +\ wrong\ faulty\ +\ wrong\ not}$		$\frac{128}{128\ +\ 11\ +\ 15}$	83.12%
accuracy	$\frac{all\ correct}{all\ correct\ +\ all\ wrong}$		$\frac{128\ +\ 4{,}730}{128\ +\ 4{,}730\ +\ 116\ +\ 11\ +\ 15}$	97.16%

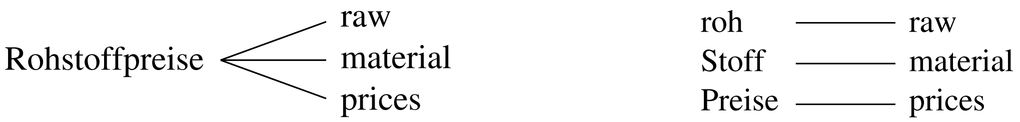
Table 8.2.: Evaluation terminology and metrics, taken from (Koehn and Knight, 2003), enrichted with examples. **correct split** = should be split and was split correctly, **correct not** = should not be split and was not, **wrong split** = should not be split but was split, **wrong not** = should be split but was not, **wrong faulty** = should be split but was split wrongly.

8.1.2. Evaluation Metrics

In the following, we describe the evaluation metrics we will use throughout this chapter to measure the accuracy of the compound splitting approaches on the gold standards. They correspond to the concepts of precision and recall, which have their origin in Information Retrieval. Nowadays they are widely used across different NLP applications. We adapt the metrics and their terminology from (Koehn and Knight, 2003), who customized the formulas for compound decomposition, see Table 8.2 for details.

Moreover, we give illustrative examples in Table 8.2: e.g. *wrong faulty*[31] denotes splittings of the system where a splitting was desired, but the two splittings do not match, e.g. the compound *"Goldbarren"* (= "gold ingot") should be split into *"Gold"* (= "gold") and *"Barren"* (= "ingot"), but the system split it into *"Gold|Bar|Ren"* (= "gold|bar|reindeer") instead. Besides category examples, Table 8.2 also features an example calculation for *precision*, *recall* and *accuracy*.

[31] This terminology is adopted from Koehn and Knight (2003).



(a) 1:n correspondence

(b) 1:1 correspondence

Figure 8.1.: Example of how compound splitting may lead from a 1:n correspondence between source and target language to a 1:1 correspondences.

8.2. Translational Correspondences Gold Standard

In this section, we introduce a gold standard that incorporates translational knowledge.[32] Recall that one of the intuitive ideas behind compound processing for statistical machine translation is to enhance the number of one-to-one correspondences between a compounding language and a non-compounding language through splitting compounds into their component words prior to training (see also Section 5.1.2).

The translation correspondence gold standard we use is based on this intuition. It is comparable to the one-to-one correspondence gold standard of (Koehn and Knight, 2003). Only compounds that have in fact been translated compositionally into two or more words of the target language are annotated. The annotation is thus not dependent on compositionality (or other linguistically motivated) assumptions of the annotator. The performance of the compound splitting approaches on this gold standard thus approximates the effect the splitting will have in end-to-end SMT.

8.2.1. Annotation Details

Here, we will present some details concerning the creation of the translational correspondences gold standard. We started from the test set of the 2009 workshop on statistical machine translation,[33] for which human reference translations are available. We took the first 5,000 words of the German testset (*news-dev2009b*), and manually annotated compound splits wherever a compound was aligned to more than one corresponding word in the English reference translation. Corresponding words were identified by the human annotator through reading the whole English sentence. In the course of this gold standard annotation, all tasks were performed manually, i.e. no statistically determined word alignment(s) were used.

[32] In Fritzinger and Fraser (2010) this gold standard is called one-to-one correspondence standard.

[33] `http://www.statmt.org/wmt09/translation-task.html`

Nr.	German compound	English reference translation	Split decision
1	Roh|Stoff|Preise ***raw|material|prices***	**raw|material|prices**	split
2	Speise|Öl *food|**oil***	vegetable **oil**	split
3	Pfand|Flasche ***deposit|bottle***	**bottle** with a **deposit** on	split
4	Regierung|Wechsel ***government**|change*	put a new **government** in office	no split
5	Preis|Steigerung ***price**|increase*	**prices** have still gone up	no split
6	Nachbar|Staaten ***neighbour**|states*	**neighbours**	no split

Table 8.3.: Examples of German compounds and their translational correspondences taken from human reference translations. Literal translations of component words are given beneath the compounds. The "split decision" column indicates whether or not phenomena of this kind were annotated as compounds to be split in the **translational correspondence gold standard**.

Most German compounds have a transparent, compositional semantics. For those, the annotation of split points is straightforward. Consider for example *"Rohstoffpreise"* (= "raw material prices") in Figure 8.1, where the German compound is not only compositional, but also has a semantically and structurally equivalent counterpart in the English reference. However, this need not always be the case. We were faced with a number of less straightforward cases, for which we give examples in Table 8.3. We distinguish the following phenomena (numbers in brackets refer to numbering in Table 8.3):

Exact Counterpart Missing in English Language (2) For example, there are German compounds (e.g. *"Speise|Öl"*) which have a transparent, compositional semantics (literally: *food|oil* = oil which is suitable for the preparation of food), but the semantics of their English compound counterpart describing the same product (*vegetable oil*) is slightly different. Whenever no exactly matching counterpart is available in English, but the correct (compositional) translation is found in the reference sentence, the compound is annotated as to be split: at least, splitting enhances the number of correct alignments between parts of the compounds (here: *"Öl"* - "oil") and the literal translation of the other word part(s) is often semantically related to the meaning of the whole compound.

German: Compound, English: Noun Phrase (3) It is not surprising that German compounds were often translated into English noun phrases instead of English compounds. If the correspondences of the German compound's component words (e.g. *"Pfand|Flasche"* = lit.: "deposit|bottle") were found in the English noun phrase construction of the reference sentence (e.g. "bottle with a deposit"), the compound is annotated as to be split: most probably, splitting will lead to correct word alignments between the former compound parts and the nouns of the English noun phrase.

Diverging Translation (4+5) In contrast, consider the case of *"Regierung|Wechsel"* (= lit.: "government|change"), where an exactly matching English counterpart of the German component parts exist (*"Wechsel"*="change"), but which is not to be found in the reference translation ("new government" instead of "government change"), we decided not to annotate the compound. Similarly, compounds like *"Preis|Steigerung"* (= lit.: "price|increase"), are **not** annotated as compounds to be split, if the structure and lexical choice deviates too much from the literal translation of the component parts (here: "increase" vs. "have still gone up"). In such cases, it is unlikely that a splitting would lead to improved word alignments.

Missing Component Translation (6) Finally, given a German compound like *"Nachbar| Staaten"* (= lit.: "neighbour|states"), we often observed a loss of information in the English reference sentence (*"Nachbarstaaten"* → "neighbours"). In lack of a counterpart for each of the components, the German compound was **not** annotated as to be split. Here, splitting might lead to a n:1 word alignment, which is less desirable than the 1:1 alignment between the whole compound and the corresponding English word.

Following these annotation criteria, 149 compounds with one-to-one translational correspondences were annotated among the 5,000 first words of the testset from the 2009 wmt shared task on statistical machine translation.

In the course of this gold standard annotation, we found several cases of lexicalised German compounds which are translated as one word in English and are thus not annotated, e.g. *"Handschuh"* (= "glove", lit. "hand|shoe"). Apart from them, we also found 3 semantically opaque compounds among the 5,000 words of the testset, namely *"Zwickmühle"* (= "catch-22 situation", lit. "tweak|mill"), *"Dunkelziffer"* (= "estimated number of unknown cases", lit. "dark|figure") and *"Tageslicht"* within the collocation *"ans Tageslicht*

splitting	**Correct**		**Wrong**			**Metrics**		
↓ approach	**split**	**not**	**split**	**not**	**faulty**	**precision**	**recall**	**accuracy**
no split	0	4,853	0	147	0	0%	97.06%	97.06%
basic freq.	75	4,380	466	**11**	68	12.32%	48.70%	89.10%
extended freq.	85	4,624	222	16	53	23.61%	55.19%	94.18%
POS	92	4,730	116	32	30	38.66%	59.74%	96.44%
Smor	122	4,664	182	**11**	21	37.54%	79.22%	95.72%
Smor -d	**128**	4,730	116	15	**11**	50.20%	**83.12%**	97.16%
Smor -d NN	121	**4,773**	**73**	22	**11**	**59.02%**	78.57%	**97.88%**

Table 8.4.: Accuracies of the different splitting approaches with respect to the **translational correspondences** gold standard. The best numeric scores per column are **bold** faced.

kommen" which is used almost identically in English (= "to come to light", lit. "to daylight come"). This observation confirms our initial assumption that most of the German compounds are semantically transparent.

8.2.2. Results

The accuracies of the different splitting approaches measured with the evaluation metrics as presented in Section 8.1.2 above are given in Table 8.4.[34] As a baseline, we indicate the accuracy of not splitting compounds at all (*no split*). Note that one of the characteristics of this gold standard is a low number of compounds to be split. In fact, of the 5,000 words, only 149 are compound words to be split, i.e. only 2.98%. As a consequence, the *no split* baseline reaches a high accuracy of 97.02%. To yield competitively high accuracies, the splitting approaches must thus not only split compounds accurately, but more importantly, words that should not be split must remain unsplit.

For the extended frequency-based approach, we calculated scores for different combinations of minimal part size and minimal part frequency (see Table B.1 in Appendix B.1 for details) and found a minimal part size of 4 characters[35] and a minimal part frequency of 3 to give reasonable precision and recall scores.

It can be seen from Table 8.4 that the two frequency-based approaches only split about half of the compounds (namely *correct split:* 75 and 85 of 149, respectively) correctly. On the other hand, they heavily over-split words that should not have been split in the

[34] In Fritzinger and Fraser (2010), we published a similar evaluation on the same data set.

[35] This is in contrast to Stymne (2008), who used 3 characters, but in accordance with Durgar El-Kahlout and Yvon (2010), who also used 4 characters.

first place (*wrong split:* 466 and 222). The more linguistic knowledge (in terms of POS restrictions or full morphological analysis) is added, the less over-splitting effects are observable, except for the deep analysis version of Smor (without *-d*, *wrong split:* 222).

The comparison of the approaches with *no split* (*accuracy:* 97.02%) shows that only the two Smor *-d* approaches are able to scarcely outperform this baseline (Smor *-d*: 97.16%, Smor *-d NN*: 97.88%). It can be seen that Smor *-d* yields the most correct splittings (128), but nevertheless, Smor *-d NN* scores highest in accuracy, because it leaves more words unsplit (*correct not:* 4,773 vs. 4,730). However, despite Smor *-d NN* reaching the highest overall accuracy score, the comparison to Smor *-d* shows that this goes at the expense of recall (Smor *-d*: 83.13% vs. Smor *-d NN*: 78.57%). Summing up, we can say that Smor *-d* yields a reasonable balance between precision and recall score and is thus considered to perform best on this translational correspondence gold standard.

Concluding Remarks The underlying idea of the translational correspondences gold standard is intuitively clear: compound splitting enhances the number of one-to-one alignments and thereby improves word alignment and translation quality. By measuring how many of the German compounds that have been translated into two or more English content words (by a human translator) are correctly split by the different approaches, we theoretically can estimate which of the approaches will have most positive impact on translation quality. However, Koehn and Knight (2003) showed, that the best performing splitting procedure on the translational correspondences standard is not necessarily the best performing splitting approach in terms of translation quality in end-to-end SMT.

A major drawback of this gold standard is its reliance on human reference translations. Apart from their potential lack of availability, human translators (or even one and the same translator) might not always translate a German compound consistently throughout the whole text. In addition to that, when using former datasets from the workshop for statistical machine translation, it might make a difference whether German was the source or the target language at the time of human translation.

This type of gold standard is a little more time-consuming than others with regard to the fact that a human annotator must take both the source and the target language sentence into consideration for his splitting decision. However, this workload can easily be distributed over different compound annotators without requiring any deeper linguistic instructions, as the splitting decision is not dependent on the compositionality of a compound but simply taken from the human reference translations.

As the annotation happens on token-level, there might not be many compounds in the gold standard in the end (as in our case: 149 compounds in a set of 5,000 words), and the evaluation of approaches against this gold standard is then biased to precision with respect to leaving words unsplit, instead of favouring an approach which has high precision on how compounds are split. In the following sections, we thus report on two additional gold standard evaluations that have been created on type-level, without reliance to human reference translations.

8.3. Linguistic Gold Standard

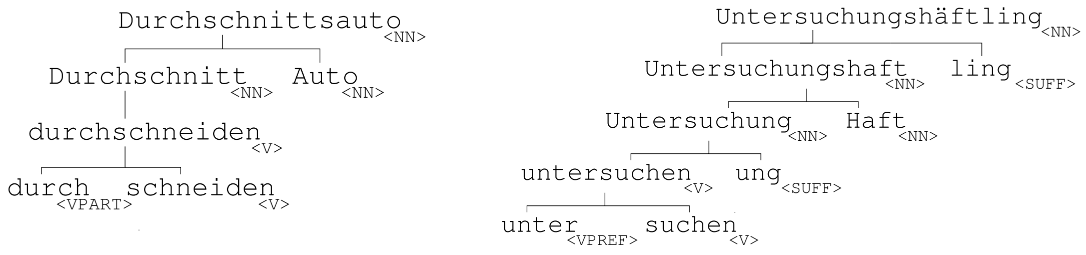
This gold standard is build from a unique word list. It differs from the previous gold standard in Section 8.2 above in that compounds are annotated on type-level and translational knowledge was **not** taken into account. Instead, the annotator considers the compositionality of a compound. To give an example, the compositional German compound *"lebenswichtig"* is split into *"Leben|wichtig"* (= "life|important"), even though its English translation consists of only one word, "vital", which means that it would not have been split in the translational correspondence gold standard.

8.3.1. Annotation Details

We start from the development set of the 2007 workshop on statistical machine translation.[36] The data set is first tokenised (using the shared task tokeniser) and then truecased by keeping the most frequent casing variant of each word. After tokenisation, the data set consists of 26,087 word forms (tokens) of 6,360 different types. For these 6,360 different types, we annotated the most plausible splits into word stems or particles. Only word formation is annotated. Derivational processes like e.g. suffixation or prefixation are ignored. Compound heads remain in the word form they occurred, they are not lemmatised in the course of the annotation process. In the following, we give some more details sorted by the phenomena they concern.

Analysis Depth The gold standard should contain plausible split points of compounds. In most cases, the highest morphological analysis level is sufficient to identify the compo-

[36] `http://www.statmt.org/wmt07/shared-task.html`



(a) *"Durchschnittsauto"* (= "average car") is annotated to be split into *"Durchschnitt|Auto"*.

(b) The structure of *"Untersuchungshäftling"* (= "person being imprisoned on remand") reveals that it should not split into *"Untersuchung|Häftling"* (= "investigation|prisoner").

Figure 8.2.: Deep morphological analyses help the decision of annotating split points on the highest analysis level.

nent words of a compound, see e.g. *"Durchschnittsauto"* (= "average car") in Figure 8.2 (a). In contrast, considering the deep morphological analysis of e.g. *"Untersuchungshäftling"* (= "person being imprisoned on remand") in Figure 8.2 (b) indicates that this compound should not be split into *"Untersuchung|Häftling"* (= "investigation|prisoner"), as the nominal suffix *"-ling"* does not attach to *"Haft"*, but to *"Untersuchungshaft"* (= "investigative custody"). Splitting the compound leads to a shift in the semantics of the compound *"Untersuchungshäftling"*: it would lead to a "prisoner under investigation" but the original meaning denotes a "person being in investigative custody". These kinds of compounds thus remain unsplit in the gold standard.

Reduction of Compound Modifiers In contrast to compound heads, all compound modifiers are reduced to their lemma form: all morphological operations that were required for putting words together are reversed. That applies to nouns, e.g. *"Düne**n**sand"* (= "dune sand", lit. "dunes sand") is split into *"Düne|Sand"*, but also to other parts of speech, e.g. verbs as in *"Mietangelegenheiten"* (= "rental matters") which is split into *"miet**en**|Angelegenheit"* (= "to rent|matters").

Hyphenated Words The words in modifier position of hyphenated German compounds are often either foreign words (*"Babyboom-Generation"* = "baby boom generation"), named entities (*"Kyoto-Protokoll"* = "kyoto protocol") or acronyms (*"EU-Mitglieder"* = *"member of the EU"*). In general, they are split at the hyphens and also within the words, if applicable. To give an example, the hyphenated compound

"Tsunami-Frühwarnsystem" (= "tsunami early warning system") is split into *"Tsunami|früh|warnen|System"* (= "tsunami|early|to warn|system").

Foreign Language Material Many foreign language words that are used in current German language are compounds in the original language themselves. For the gold standard, we allow foreign language words to be component words of German compounds. In contrast, split points within foreign language words themselves are not annotated, e.g. *"Fund-rasingverband"* (= "fundraising association") is split into *"Fundraising|Verband"*. There is not splitting into *"fund|raising"*.

Particle Verbs German particle verbs allow for an optional separation of the particle, depending on word order and tense. In the gold standard, we allow for particles to be split only if these can be used separately from the verb in a grammatically sound sentence:[37] we split *"aufgibt"* (= "give up") into *"auf|gibt"*, but we leave the past participle form *"aufgegeben"* (= "given up") unsplit, as the latter one (*"*gegeben"*) cannot occur separately.[38]

Following these annotation criteria, we found 1,100 compound types among the 6,187 words of the gold standard. Not surprisingly, most of them are noun compounds (806). Among the others, we found 201 particle verbs and 93 adjectives.

8.3.2. Results

We measured the accuracies of the different splitting approaches with the evaluation metrics presented in Section 8.1.2 above. The results on the linguistic gold standard are given in Table 8.5.[39] As a baseline, we indicate the accuracy of not splitting compounds at all (*no split*). Note that for the extended frequency-based approach, we calculated scores for different combinations of minimal part size and minimal part frequency (which we present in Table B.2 in Appendix B.2). However, here, we show only the results for minimal part size of 4 characters and minimal part frequency of 3, as these yielded reasonably balanced precision and recall scores.

Generally speaking, the performance of all splitting approaches on this linguistic gold standard is similar to their performance on the translational correspondences gold stan-

[37] See Section 7.1.5 for more details on when to split German particle verbs.

[38] See Figure 7.6 on page 84 for an illustrative example.

[39] Note that we published a similar evaluation on the same data set in Fritzinger and Fraser (2010).

splitting	**Correct**		**Wrong**			**Metrics**		
↓ approach	**split**	**not**	**split**	**not**	**faulty**	**precision**	**recall**	**accuracy**
no split	0	5,087	0	1,100	0	0%	82.22%	82.22%
basic freq.	575	3,976	1,112	107	417	27.33%	52.32%	73.56%
extended freq.	623	4,584	504	255	221	46.22%	56.69%	84.16%
POS	661	4,933	155	369	69	74.69%	60.15%	90.42%
Smor	**990**	4,895	193	**32**	77	78.57%	**90.08%**	95.12%
Smor -d	917	5,037	51	116	66	88.68%	83.44%	**96.23%**
Smor -d NN	677	**5,056**	**32**	373	**49**	**89.31%**	61.60%	92.66%

Table 8.5.: Accuracies of the different splitting approaches with respect to the **linguistic** gold standard. The best numeric scores per column are **bold** faced.

dard (reported in Section 8.2.2 above). While the frequency-based approaches suffer from poor precision which is due to heavy over-splitting (e.g. 1,112 *wrong split* for the *basic frequency* approach), both the precision and the overall performance rises, the more linguistic knowledge (POS, Smor) is added to the approaches. Overall, the Smor *-d* splittings fit the gold standard best, with an accuracy of 96.23%, despite the fact that Smor reaches higher recall (90.08% vs. 83.44%, and also most *correct splits*, namely 990) and Smor *-d NN* reaches higher precision (89.31% vs. 88.68%, mainly due to its high *correct not split* score of 5,056). Smor is the only splitting approach that scores reasonably high in both precision (88.68%) and recall (83.44%).

8.4. External Domain-Specific Gold Standard

In addition to the hand crafted gold standard described in the previous Section 8.3, we used a freely available external gold standard for noun compounds (Marek, 2006)[40] to evaluate the accuracy of our splitting procedure(s). This gold standard comes as a by-product of the developement of a weighted finite-state transducer for German nominal compounds was developed. Due to the fact that our compound splitting approach is based on a finite-state based morphological analyser (Smor) that is similar to the one developed by Marek (2006) makes this a suitable dataset for external evaluation.

	word	**gold annotation**	**gloss**	**details**
1	Risikopotenzial	Risiko{N}+Potenzial{N}	risk potential	split points are marked "+"
2	Spinnenfinger	Spinne|n{N}+Finger{N,V}	spider fingers	inserted letters are marked "|"
3	Farbbild	Farb**,**e{N}+Bild{N,V}	color picture	deleted letters are marked "**,**"
4	Mediengestalter	Medi,um|en{N}+Gestalter{N}	media designer	substitutions: deletion and filler letter
5	Kultfiguren	Kult{N}+Figur**(**en**)**{N}	cult figurs	inflectional endings are marked "**()**"
6	Obstgärten	Obst{N}+G**A**rten{N}	fruit orchard	umlautung phenomena are marked with capital letters (here: "**A**")

Table 8.6.: Featured annotations of the external gold standard.

8.4.1. Annotation Details

Data/Creation The external domain-specific gold standard is based on data from a German computer magazine for semi-professional computer users, *c't*[41], which appears bi-weekly. All texts from the issues 01/2000 to 13/2004 were used, in total 117 magazines, adding up to 20,000 pages of A4 text (= 15 million tokens). After filtering out lower-cased[42] and function words (using STTS' list of closed word class members[43]) from this text collection, 378,846 words remained. Among them were compound and simple words, nouns and words of other word classes. A word list derived from a German lexicon,[44] including additional hand-crafted entries by Marek (2006) was then used to filter out words that were neither compositional nor nouns. This procedure resulted in a list of 158,653 nominal compound words. The gold standard annotation was performed semi-automatically by using a simple compound splitter (similar to the one described in Koehn and Knight, 2003), that was asking for human advice in the (around 12,000) cases of doubt that occurred. All errors that Marek (2006) detected while further developing the weighted FST were fixed, so that in the end, the number of erroneous annotations was estimated to be about 3%.

[40] `http://diotavelli.net/files/ccorpus.txt`

[41] `http://www.heise.de/ct`

[42] The gold standard was designed for nominal compounds and German nouns are always upper-cased.

[43] = The **S**tuttgart **T**übingen **T**ag **S**et, created 1995/99, cf. `http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-table.html`.

[44] CELEX-2, release 2.5

	gold standard format	**our format**	**format adaptation**
1	Risiko{N}+potenzial{N}	Risiko Potenzial	substitute "+" with whitespace
2	Spinne|n{N}+Finger{N,V}	Spinne Finger	remove inserted letters
3	Farb,e{N}+Bild{N,V}	Farbe Bild	re-introduce deleted letters
4	Medi,um|en{N}+Gestalter{N}	Medium Gestalter	use original word stem
5	Kult{N}+Figur(en){N}	Kult Figuren	keep inflectional endings
6	Obst{N}+GArten{N}	Obst Gärten	keep umlautung

Table 8.7.: Required formatting adaptations for the external gold standard.

Featured Annotations Besides structural annotations (i.e. split points), the external gold standard features some additional annotations: each word and word part is annotated with its word class(es): for example, "**{N}**" indicates nouns (cf. *potenzial{N}*, Table 8.6, row 1), while "**{N,V}**" indicates that the word (part) is either a noun or a verb (cf. *bild{N, V}* Table 8.6, row 3). Moreover, it indicates the required transformations from simple nouns into compound modifiers (and vice versa); cf. Table 8.6, rows 2-4: insertions: *Spinne* → *Spinne**n***, deletions: *Farb**e*** → *Farb*, substitutions: *Medi**um*** → *Medi**en***). Finally, the external gold standard annotation includes base forms and inflectional endings (cf. Table 8.6, rows 5 and 6).

Required Adaptations The external gold standard features some annotations that either do not match the output of our compound splitter or they are not relevant to the evaluation of splitting accuracy: e.g. reduction to the base word form (= removal of inflectional endings) belongs to the latter of these two categories. Table 8.7 illustrates (minor) format adaptations that were performed. Furthermore, we re-merged split particles in the external gold standard, as we allow separated particles only for verbs:

word	**gloss**	**gold standard format**	**after modification**
Hauptaufgabe	main task	Haupt{N}+auf{PREP}+Gabe{N}	Haupt{N}+Aufgabe{N}

Finally, verbs occurring in modifier positions are kept in a shortened stem representation (without any inflectional ending) in the external gold standard. In contrast, our format represents such verbs as lemmas (with infinitive form ending). We solved this by adding the regular German infinitive ending *"en"* to all modifying verbs, ignoring the very few phonotactically driven exceptions:

word	**gloss**	**gold standard format**	**after modification**
Sprengstoffe	explosive substances	spreng{V}+Stoff(e){N}	spreng**en**{V}+Stoffe{N}

splitting	**Correct**		**Wrong**			**Metrics**		
↓ approach	**split**	**not**	**split**	**not**	**faulty**	**precision**	**recall**	**accuracy**
no split	0	4,408	0	154,245	0	0	2.78%	2.78%
basic freq.	84.924	1,006	3,402	**1,439**	67,882	54.37%	55.06%	54.16%
extended freq.	102,824	2,823	1,585	12,370	39,051	71.67%	66.66%	66.59%
POS	122,553	3,971	437	15,369	16,323	87.97%	79.45%	79.75%
Smor	**135,490**	4,112	296	5,681	**13,074**	**91.02%**	**87.84%**	**87.99%**
Smor -d	131,177	4,171	237	7,811	15,257	89.44%	85.04%	85.31%
Smor -d NN	130,838	**4,190**	**218**	8,174	15,233	89.44%	84.82%	85.11%

Table 8.8.: Accuracies of the different splitting approaches with respect to the **external domain-specific** gold standard. This consists of 158,653 nouns, whereof 154,245 are to be split. The best numeric scores per column are **bold** faced.

This applies only to modifiers that were assigned only {V} (like in the *"sprengstoff"* example). Whenever the modifying tag was ambiguous, we chose not to treat the modifier as verb and thus no infinitive ending was added: e.g. bau{V,N}+arbeiter{N} → bau{N}+arbeiter{N} (construction worker).

8.4.2. Results

Even for the external gold standard, we calculated the accuracies of the different splitting approaches using the evaluation metrics as presented in Section 8.1.2 above. The accuracies on the external gold standard are given in Table 8.8. For this gold standard evaluation, we did bit compare different parameter settings of the extended frequency-based approach. We give the results for minimal part size of 4 characters and minimal part frequency of 3, as these yielded reasonably balanced precision and recall scores in the two previous gold standard evaluations. Despite its different characteristics (in terms of size, compound density and domain), the performance of the different splitting approaches on this gold standard deviates only slightly from the other gold standards.

It can be seen from Table 8.8 that Smor splits 135,490 of the 154,245 compounds correctly (87.84%), which is roughly 30% more than the basic frequency baseline yields (84,924 of 154,245, corresponding to 55.05%). Again, the splittings of Smor *-d NN* are most conservative in that most words are left *correct not split*, namely 4,190 and least words are *wrong split* (218). Moreover noticeable is the high number of *wrong not split* words of the *POS*-based splitting approach: 15,369, which we attribute to the high number of domain-specific (here: technical) single word terms that have not occured in the lexicon which was used for filtering the data during pre-processing (see Section 6.3.1 for details). In terms of overall accuracy, Smor performs best on the external domain-specific gold standard, compared to Smor *-d NN* which performed best on the translational correspondences standard and Smor *-d* which performed best on the linguistic gold standard.

Translational Correspondence Gold Standard Results								
splitting	**Correct**		**Wrong**			**Metrics**		
↓ approach	**split**	**not**	**split**	**not**	**faulty**	**precision**	**recall**	**accuracy**
no split	0	4,851	0	149	0	–	–	97.02%
basic freq.	75	4,380	466	**11**	68	12.32%	48.70%	89.10%
extended freq.	85	4,624	222	16	53	23.61%	55.19%	94.18%
POS	92	4,730	116	32	30	38.66%	59.74%	96.44%
Smor	122	4,664	182	**11**	21	37.54%	79.22%	95.72%
Smor -d	**128**	4,730	116	15	**11**	50.20%	**83.12%**	97.16%
Smor -d NN	121	**4,773**	**73**	22	**11**	**59.02%**	78.57%	**97.88%**

Linguistic Gold Standard Results								
splitting	**Correct**		**Wrong**			**Metrics**		
↓ approach	**split**	**not**	**split**	**not**	**faulty**	**precision**	**recall**	**accuracy**
no split	0	5,088	0	1,099	0	-	0.00%	82.23%
basic freq.	577	3,984	1,105	99	422	27.42%	52.55%	73.72%
extended freq.	634	4,598	491	241	223	47.03%	57.74%	84.56%
POS	656	4,935	154	367	75	74.12%	59.74%	90.37%
Smor	**967**	4,877	212	**50**	81	76.75%	**88.07%**	94.46%
Smor -d	894	5,018	71	135	69	86.46%	81.42%	**95.56%**
Smor -d NN	671	**5,054**	**35**	375	**52**	**88.52%**	61.11%	92.53%

External Domain-Specific Gold Standard Results								
splitting	**Correct**		**Wrong**			**Metrics**		
↓ approach	**split**	**not**	**split**	**not**	**faulty**	**precision**	**recall**	**accuracy**
no split	0	4,408	0	154,245	0	0	2.78%	2.78%
basic freq.	84.924	1,006	3,402	**1,439**	67,882	54.37%	55.06%	54.16%
extended freq.	102,824	2,823	1,585	12,370	39,051	71.67%	66.66%	66.59%
POS	122,553	3,971	437	15,369	16,323	87.97%	79.45%	79.75%
Smor	**135,490**	4,112	296	5,681	**13,074**	**91.02%**	**87.84%**	**87.99%**
Smor -d	131,177	4,171	237	7,811	15,257	89.44%	85.04%	85.31%
Smor -d NN	130,838	**4,190**	**218**	8,174	15,233	89.44%	84.82%	85.11%

Table 8.9.: Accuracies of the different splitting approaches with respect to the different gold standard standards. The best numeric scores per column (separately for each gold standard) are **bold** faced.

8.5. Comparison of Results

To sum up, out of the three gold standards we used, the external domain-specific gold standard is the most challenging one. On the one hand, it is much larger than the other ones, and it also exhibits a very high density of compounds. It is thus highly predictive with respect to correct compound splittings, i.e. **how** compounds are split.

Moreover, the linguistics-based splitting approaches, *POS* (using TreeTagger) and also all Smor approaches, are potentially faced with out-of-vocabulary words due to the domain-specific technical terminology. Nevertheless, all approaches performed similarly on this gold standard as they did on the translational correspondences and the linguistic gold standards. In Table 8.9, we present an overview of the performances of all splitting approaches across all gold standards we investigated. The results of each gold standard are repeated and compiled into one single table for the purpose of a better overview. It can be seen that he morphologically-aware compound splitting clearly outperforms the other approaches. However, the performances of the three Smor-based splitting variants differ across gold standards. We will thus keep all splitting approaches that were under investigation here and evaluate their performance in end-to-end SMT evaluation in Chapter 9. But before that, we discuss typical errors we revealed in the course of the gold standard evaluations for each of the splitting approaches in the next section.

8.6. Error Analysis

In the course of the gold standard evaluations, we discovered several recurrent errors for each of the different splitting approaches, some of which we want to illustrate with examples in this section. For each splitting approach – frequency-based, POS-based and Smor-based – we give some examples, which are grouped according to their error causes. Section 8.6.1 starts with an error analysis of the frequency-based approach, followed by Section 8.6.2 with typical errors of the POS-based approach. In Section 8.6.3, we give illustrating example errors, and a detailed analysis of all errors that occured in the Smor -d splitting approach.

word	**basic frequency-based splitting**	**high-frequent word**
bombardieren *to bomb*	Bom|Bar|die|Ren **Bom|bar|the|reindeer*	article *"die"* = "the"
Ausschreitungen *riots*	aus|Schrei|tun|Gen *out|scream|to do|gene*	preposition *"aus"* = "out"
verantwortlich *responsible*	Vera|two|RTL|ich *Vera|two|RTL|I*	pronoun *"ich"* = "I"

Table 8.10.: Typical errors of the **basic** frequency-based approach (Koehn and Knight, 2003), due to high-frequent function words, pronouns and acronyms.

8.6.1. Errors of the Frequency-based Approach

Recall from Section 6.2 above that the frequency-based splitting does not include any kind of linguistic knowlege, except for a small set of filler letters and a manually compiled stop list. It is thus not surprising that a purely substring- and frequency-based approach leads to word splittings that are **not** linguistically motivated.

Basic vs. Extended Approach The *basic* frequency-based splitting approach is our reimplementation of the original algorithm as first described by Koehn and Knight (2003), see Section 6.2.1 for details. In this approach, the minimal word part size is set to three characters, only filler letters *"s"* and *"es"* and deletion letter *"n"* are allowed and no stop list is used. As a consequence, the *basic* approach leads to countless erroneous splits into high-frequent German words such as articles (e.g. *"die"* = "the" in *"bombardieren"*, cf. Table 8.10), prepositions (e.g. *"aus"* = "out" in *"Ausschreitungen"*) or pronouns (e.g. *"ich"* = "I" in *"verantwortlich"*, cf. Table 8.10), which usually cannot be part of a sound German compound. However, as most of these errors are uninteresting from a linguistic viewpoint, the error analysis of the remainder of this section will focus on errors of the extended frequency-based approach, as described in Section 6.2.2.

Word Part Frequency-driven Splitting Errors This group of errors comprises cases where the correct splitting was among the splitting options, but due to high word part frequencies of another option, an erroneous splitting scored highest and was picked. Some examples of such frequency-driven splitting errors are given in Table 8.11,[45] which

[45] We will here have a closer look at errors of the extended frequency-based approach, but obviously, the example errors given in Table 8.10 for the basic frequency-based approach also fall into this category.

Nr.	word	frequency-based splitting	correct splitting
1	Mitverantwortlichkeit *co-responsibility*	mitverantwortlich|keit *co-responsable*|*ility*	- no splitting -
	Nachwuchs *the offspring*	nach|wuchs *after*|*grow*	- no splitting -
2	dramatisch *dramatic*	Drama|Tisch *drama*|*table*	- no splitting -
	beigesteuert *contributed*	Beige|steuert $beige_N$|$controls_V$	- no splitting -
3	Werkstattleiter *workshop manager*	Werk|statt|Leiter *factory*|*instead_ of*|*manager*	Werkstatt|Leiter *workshop*|*manager*
	Baustoffen *building materials*	baust|offen $build_{V,2ndperson}$|$open_{ADJ}$	Bau|Stoffen *construction*|*materials*

Table 8.11.: Examples for frequency-driven over-splittings or erroneous splittings: 1) ungrammatical splitting into prefixes/suffixes, 2) examples of over-splittings into standalone entities, 3) erroneous splittings: word should have been split, but differently (see last column).

is divided into three parts: i) ungrammatical splitting into prefixes/suffixes, ii) examples of over-splittings into standalone entities, iii) erroneous splittings, where the word should have been split, but differently.

The case of *"Mitverantwortlichkeit"* (= "co-responsibility") shows an over-splitting into the adjective *"mitverantwortlich"* and the nominal suffix -*"keit"*. In German, such nominal suffixes cannot occur separated from their noun. It probably occurred in the word part frequency corpus as a by-product of hyphenation or bad tokenisation.[46] The example of *"Nachwuchs"* (= "the offspring") is similar, even though here, the verbal particle *"nach"* (= "after") can very well be separated from its verb (as in e.g. *"nach|wachsen"* = "to grow again") but not after the nominalisation of the particle verb into *"Nachwuchs"*.

In the second group of Table 8.11, we give examples for erroneous splittings into standalone words, that are nevertheless not sound from a linguistic point of view. The adjective *"dramatisch"* (= "dramatic") cannot be split into the two nouns *"Drama"* (= "drama") and *"Tisch"* (= "Tisch"). Note however, that casing plays an important role here: if the word had appeared upper-cased, and thus be used as a noun, it could – theoretically and despite being semantically implausible – be split into exactly these two

[46] Note that without the minimal part size constraint of 4 characters, this word would also have been split into the high-frequent German preposition *"mit"* = "with".

words. In contrast, that is not possible for the case of *"beigesteuert"* (= "contributed"), which is a verb participle that could never be split into the colour "beige" and the verb *"steuert"* (= "to control", in 3rd Person).

In contrast, the third group of errors as given in Table 8.11 contains true compounds that should have been split, and in fact were split, but wrongly. These are examples of the category *wrong faulty*, as given in the results tables of the different gold standard evaluations. The compound *"Werkstattleiter"* ("workshop manager") contains two split points of which the first one into *"Werk"* and *"statt"* is wrong, but, at least, the head noun *"Leiter"* (= "leader, manager") is correctly identified. It is typical for the frequency-based approach to split into too many parts, where possible. This happens because higher frequent words are often shorter (as is the case here for the function word *"statt"* = "instead"). Another category of errors are so-called parasite words (Airio, 2006). For example, there are two possible (correct) ways to split *"Gastraum"* depending on the context in which it occurs: either *"Gas|Traum"* = "gas|dream" or *Gast|Raum* = "guest|room"), where usually one option is strongly preferred. The last example of Table 8.11 *"Baustoffen"* (= "building materials") also falls into this category, even though here, only one of the two options is linguistically sound according to the composition rules of German. A noun can never be split into a finite verb *"baust"* (= "you build") and an adjective *"offen"* = ("open").

To conclude, the word part frequency-driven errors we discussed here are problematic for all approaches under investigation (frequency, POS and morphologically-aware), as they do not concern the creation of different splitting options, but the disambiguation of splitting options. The reason why the frequency-based approach often favours such erroneous splittings is that – due to having fewer constraints – more *poor* splitting options are available prior to the frequency-driven disambiguation routine, and thus more erroneous splittings come out in the end.

Transformation-based Errors This kind of error describes cases where the stripping of filler letters and/or the addition of deletable letters leads to splittings into unplausible word parts. Note that in contrast to the frequency-driven errors of the previous paragraph, this kind of error hardly ever occurs in linguistically well-informed approaches due to their inherent knowledge about stems and possible morphological operations for word formation.

Nr.	word	frequency-based splitting	correct splitting
4	steigern *increase*	Stein|gern *stone|gladly*	- no splitting -
5	Handlungsebenen *action level*	Hand|Lunge|Ebenen *hand|lung|level*	Handlung|Ebenen *action|level*
6	Damenstrümpfe *women's stockings*	Damen|Trümpfe *women|trumps*	Dame|Strümpfe *woman|stockings*

Table 8.12.: Examples for transformation-driven errors: 4: deletion letter "n" was erroneously identified, 5: deletion letter "e" was erroneously identified, but filler letter "s" was correct, 6: filler letter "s" was erroneously identified.

We give some examples for typical transformation-based errors in Table 8.12. As can be seen, the verb *"steigern"* (= "to increase") is erroneously split into the noun *"Stein"* (= "stone") and the adverb *"gern"* (= "gladly"). This happens because the extended frequency-based approach allows for a deletable letter *"n"* for each word that ends in a vowel. Here, the meaningless character sequence *"Stei"* is erroneously assumed to be the modifier form of *"Stei**n**"* (= "stone"), and as "stone" occurs frequently in the training data, this split is chosen.

Similarly, the example of *"Handlungsebenen"* (= "action level") shows an over-splitting of the modifier *"Handlung"* (= "action") into *"Hand|Lunge"* (= "hand|lung"). Here, the filler letter *"s"* was correctly identified, but unfortunately a deletable letter *"e"* was assumed to have been stripped the modifier for compound formation. However, if one wanted to combine the three German words *"Hand"+"Lunge"+"Ebenen"* into one compound (which is possible from a linguistic point of view, even though semantically rather implausible), the result would be *"Handlungenebenen"*, as the word *"Lunge"* does not strip its final letter for compound formation, but instead requires a filler *"n"*.

Finally, *"Damenstrümpfe"* (= "women's stockings") in Table 8.12 should have been split into *"Dame|Strümpfe"* (= "woman|stockings"), i.e. the action a splitting procedure had to perform was to identify the filler letter *"n"* which is attached to the modifier *"Dame"*. Instead, the extended frequency based approach identified an erroneous filler letter *"s"* which it assumed had been attached to the modifier *"Damen"* (= "women") and thus clipped the original head *"Strümpfe"* (= "stockings") to *"Trümpfe"* (= "trumps"). As a consequence of the fact that the plural form *"Damen"* occur more frequently in the training data than its singular *"Dame"*, the word is then erroneously split into *"Damen|Trümpfe"* (= "women|trumps").

Nr.	**word**	**frequency-based splitting**	**correct splitting**
7	Stagnieren _stagnating_	stage|Nieren _stage|kidneys_	- no splitting -
	Gaspartikel _gas particle_	gasp|Artikel _gasp|article_	Gas|Partikel _gas|particle_
8	glucksend _gurgling_	Gluck|send _*Gluck|send_	- no splitting -
	Bitebene _bit level_	bite|Bene _bite|*Bene_	Bit|Ebene _bit|level_
9	Teilchenimpuls _particle momentum_	Teil|Chen|Impuls _part|Chen|impulse_	Teilchen|Impuls _particle|impulse_
	Daumenkinoeffekt _flip book effect_	Daum|Kino|Effekt _Daum|cinema|effect_	Daumen|Kino|Effekt _thumb|cinema|effect_
10	Einsteins _Einstein's_	einst|eins _once|one_	- no splitting -
	Thatchers _Thatcher's_	that|Chers _that|*Chers_	- no splitting -

Table 8.13.: Examples of errors including foreign language material and namend entities; 7: split into English and German words, 8: split into English words and German non-sense, 9: split into proper nouns and German words, 10: split proper nouns that should not have been split.

Proper Nouns / Foreign Language Material This group of errors shows the important role of corpus cleanliness for frequency-based splitting. As the frequency-based splitting approach does not include information about well-formed stems, words are split into any substring that is found in the corpus, even into proper nouns or foreign language material. Recall that for all our experiments, we used the monolingual training corpus of the EACL 2009 workshop on statistical machine translation, which we did not pre-process or clean but took it as it is (in tokenised format). We give examples of erroneous splits into proper nouns and foreign language material based on this corpus in Table 8.13.

It may happen that German words are split into English and German words. For example *“Stagnieren”* (= “stagnating”) is split by the extended frequency-based approach into *“stage|Nieren”* (= “stage|kidneys”) or *“Gaspartikel”* (= “gas particle”) which is split into *”gasp|Artikel”* (= “gasp|article”). From a linguistic point of view, a splitting of a German compound into English and German words is only possible for a very limited number of English words that have been Germanized. These are often modern technical terms, as e.g. in *"Mailadresse"*, where a split into *"Mail|Adresse"* (= "(e-)mail|address") is valid due to the fact that *"Mail"* is used as a simple word in German, too.

Whenever German words are split into English words and German typos or non-sense words, the resulting splittings turn out to be even more absurd, as the examples of *"glucksend"* (= "gurgling") → *"Gluck/send"* (= "*Gluck|send") and *"Bitebene"* (= "bit level") → *"bite|Bene"* (= "bite|*Bene") show.

Besides the foreign language material, proper nouns are another source of errors for frequency-based splitting, particularly in cases where proper nouns are homographic to, for example, German derivational affixes. An example is *"chen"* which is a Chinese family name on the one hand, and a German derivational affix which transforms any noun into a diminutive form. For example, it makes a little bear (*"Bärchen"*) out of a bear (*"Bär"*). The same derivational process happens in the case of *"Teilchenimpuls"* (= "particle momentum"), but as German derivation affixes cannot standalone, they should not be split from the stem. *"Teil|Chen|impuls"* is thus an unplausible splitting into "part|Chen|impulse". However, this error type also occurs in cases where the proper noun is not homographic to a German affix, as the case of *"Daumenkinoeffekt"* (= "flip book effect") in Table 8.13 shows: here, the German soccer trainer Christoph Daum occurred more frequently in the corpus than *"Daumen"* (= "thumb"), which would have led to the correct splitting.

Finally, the frequency-based splitting procedure not only leads to splits of German words into proper nouns, but on the other hand also splits proper nouns into German words. An example is Albert Einstein, whose family name is split into the adverb *"einst"* (= "once") and the numeral *"eins"* (= "one"). In the case of Margaret Thatcher, the genitive form of her family name, Thatchers, was split into the English word "that" and non-sense word *"*Chers"*, which is the genitive form of the singer Cher.

All the given examples show the dependency of the frequency-based splitting approach on the corpus from which word part frequencies are derived. In contrast, the more linguistic knowledge a splitting approach incorporates, the less dependent it is on the corpus. We present some typical errors of these approaches in the following two sections.

8.6.2. Errors of the POS-based Approach

We already saw from the different gold standard evaluations that the POS-based approach obtains more accurate splittings than the two purely frequency-based approaches

Nr.	word	frequency-based splitting	POS-based splitting
+	Hierarchiestufe$_{NN}$ *hierarchy level*	hier$_{ADV}$|Archie$_{NE}$|Stufe$_{NN}$ *here|Archie|level*	Hierarchie$_{NN}$|Stufe$_{NN}$ *hierarchy|level*
	Gegenstand$_{NN}$ *object*	gegen$_{ADV}$|stand$_{vv}$ *against|stood*	Gegenstand$_{NN}$ *object*
=	Niederschlag$_{NN}$ *precipitation*	nieder$_{ADJ}$|Schlag$_{NN}$ *low|hit*	nieder$_{ADJ}$|Schlag$_{NN}$ *low|hit*
	Eisenerz$_{NN}$ *iron ore*	Eisenerz$_{NN}$ *iron ore*	Eis$_{NN}$|Erz$_{NN}$ *ice|ore*
-	Gegendruck$_{NN}$ *back pressure*	gegen$_{PREP}$|Druck$_{NN}$ *against|pressure*	Gegend$_{NN}$|Ruck$_{NN}$ *area|jerk*
	Zugriffscode *access code*	Zugriff|Code *access|code*	Zug|Riff|Code *train|reef|code*

Table 8.14.: Error examples where the POS-based approach splits + : better, = : equally good/bad, - : worse than the extended frequency-based approach.

(cf. Sections 8.2.2, 8.3.2 and 8.4.2). In the following, we give some examples which demonstrate the strengths and weaknesses of the POS-based approach in comparison to the frequency based approach. See Table 8.14 for an overview.

Better Performance The first group of examples shows that the POS-based approach often leads to better splittings and at the same time considerably reduces over-splitting. As can be seen from Table 8.14, *"Hierarchiestufe"* (= "hierarchy level") is erroneously split into the high-frequent adverb *"hier"* (= "here"), the name "Archie" and the correct head, *"Stufe"* (= "level"). As adverbs and proper nouns are not admitted to be compound parts in the POS-based approach, the splitting into *"hier"* and *"Archie"* is blocked, and the correct splitting into *"Hierarchie"* (= "hierarchy") is picked instead. The same restriction applies to *"Gegenstand"* (= "object"), where the word is left as a whole by the POS-based approach, but the frequency-based approach splits it into *"gegen"* (= "against") and *"stand"* (= "stood"). This is blocked by two POS constraints: i) adverbs do not belong to the group of valid compound parts and ii) the POS of the rightmost word part (here: the verb *"stand"*) does not match the POS of the whole compound *"Gegenstand"*, which is a noun.

Equal Performance In the examples belonging to the second group of Table 8.14, both approaches split equally well/bad: in the first case, *"Niederschlag"* (= "precipitation"), both erroneously split into the adjective *"nieder"* (= "low") and the noun *"Schlag"* (=

compound unknown Smor does not return any analysis. This group indicates lacking lexical coverage.

desired analysis missing Smor returns an analysis, but the "gold" analysis is not among the provided analyses.

lexicalised in Smor -d Smor returns the "gold" analysis, but the word is also lexicalised as a whole. The "-d" flag blocks decomposition as it outputs only the analyses with the least number of parts;

flat hierarchy The flat Smor analyses allow no conclusions about the internal hierarchy level, in contrast to the gold standard annotation. As a consequence, many words are over-split.

Figure 8.3.: Description of Smor-relared error categories.

"hit"). This splitting is wrong, even though here, all POS-restrictions are met. From a linguistic point of view, it is a nominalisation of the particle verb *"niederschlagen"* (= "to precipitate") and should thus only be split if it occurs as a verb. The second example, *"Eisenerz"* (= "iron ore") should have been split into *"Eisen"* (= "iron") and *"Erz"* (= "ore"), but unfortunately, none of the approaches gets it right: the extended frequency-based approach is blocked through the minimal part size of 4 characters and thus leaves the word as a whole. In contrast, the POS-based approach splits into *"Eis|Erz"* (= "ice|ore"), because *"Eis"* occurs more frequently in the corpus than the correct modifier *"Eisen"* does.

Worse Performance Finally, in the case of *"Gegendruck"* (= "back pressure"), both splittings are strictly speaking wrong, as this word should not have been split in the first place. However, here, the frequency-based splitting into *"gegen|Druck"* (= "against|pressure") is semantically closer to the compound than the less plausible splitting of the POS-based approach into *"Gegend|Ruck"* (= "area|jerk"). The example of *"Zugriffscode"* (= "access code") shows that in some cases, the POS-based approach splits deeper than the frequency-based approach, for example *"Zugriff"* (= "access") into *"Zug"* (= "train") and *"Riff"* (= "reef"). This erroneous split does not happen to the frequency-based approach, as there, the minimal part size is set to 4 characters.

The examples we gave in this section illustrate that POS-constraints can lead to improved splittings, but at the same time they show that even these well-defined constraints cannot prevent all erroneous splittings.

error type	**Wrong**		
	split	**not**	**faulty**
frequency-related	19	14	8
compound unknown to Smor	0	4	0
desired analysis missing in Smor	7	2	30
lexicalised in Smor -d	0	96	28
flat hierarchy	25	0	0
total number of errors	51	116	66

Table 8.15.: Smor *-d* coverage study on the **linguistic gold standard** that consists of 6,187 word types, whereof 1,100 are compounds.

8.6.3. Errors of the Smor-based Approach

In the previous paragraphs, we gave some manually selected examples for typical errors of the frequency-based and the POS-based approach, respectively. In this section, we will report on a detailed error analysis of our Smor *-d* splitting approach (cf. Sections 7.2.1 and 8.1). We manually screened all errors of this approach on the linguistic gold standard and classified them into two main error categories: i) frequency-related and ii) Smor-related errors, which we further divided into more fine-grained categories. Descriptions of these sub-categories are to be found in Figure 8.3. The whole error classification of the Smor-based approach is given in Table 8.15. In Table 8.16 the results are enriched with examples for each of these error categories. Note that the total number of errors (51 – 116 – 66) corresponds to the figures in the *Wrong* columns for Smor *-d* on the linguistic gold standard in Table 8.5 of Section 8.3 above.

Frequency-related Errors These include all splittings where the correct splitting was found among all splitting options. However, higher word part frequencies of another splitting option led to an erroneous final splitting choice. Recall that this error type occurs in all of the investigated splitting approaches as we use the same corpus-driven disambiguation strategy for all of them. These include over-splittings such as *"bleibenden"* (= "remaining") into *"bleiben|enden"* (= "to remain|ends"), erroneously not split compounds such as *"Klimawandel"* (= "climate change"), and faulty split compounds such as *"Herzinfarktrisiko"* (= "risk of heart attack"). For all of these, the frequencies of the contained word parts lead to highest scoring splits, which are erroneous.

frequency-related					
wrong: split (**19**)		wrong: not split (**14**)		wrong: faulty split (**8**)	
gold	our approach	gold	our approach	gold	our approach
bleibenden *remaining*	bleiben|Enden *remain|ends*	Klima|Wandel *clima|change*	Klimawandel *climatic change*	Herz|Infarkt|risiko *heart|attack|risk*	Herzinfarkt|Risiko *heart attack|risk*
compound unknown to Smor					
wrong: split (**0**)		wrong: not split (**4**)		wrong: faulty split (**0**)	
gold	our approach	gold	our approach	gold	our approach
n.a.	n.a.	Coca|Bauern *Coca|farmers*	Cocabauern *Coca farmers*	n.a.	n.a.
desired analysis missing in Smor					
wrong: split (**7**)		wrong: not split (**2**)		wrong: faulty split (**30**)	
gold	our approach	gold	our approach	gold	our approach
Walton *Walton*	Wal|Ton *whale|tone*	treffen|sichere *to hit|certain*	treffsichere *certain hit*	Norden|Europa *the north|Europe*	Nord|Europa *north|Europa*
lexicalised in Smor -d					
wrong: split (**0**)		wrong: not split (**96**)		wrong: faulty split (**28**)	
gold	our approach	gold	our approach	gold	our approach
n.a.	n.a.	Ziel|Gruppe *target|group*	Zielgruppe *target group*	Zentral|Bank|Chefs *central|bank|director*	Zentralbank|Chefs *central bank|director*
flat hierarchy					
wrong: split (**25**)		wrong: not split (**0**)		wrong: faulty split (**0**)	
gold	our approach	gold	our approach	gold	our approach
Schrittmacher *pacemaker*	Schritt|Macher *step|maker*	n.a.	n.a.	n.a.	n.a.

Table 8.16.: Examples for errors of the Smor *-d* splitting with respect to the error categories of in Table 8.15, to which numbers in brackets correspond to.

Compound Unknown The first group contains compounds that are unknown to Smor. As soon as one of the word parts is not covered by Smor's lexicon, these are left un-analysed and thus un-split. A typical example is *"Cocabauern"* (= "Coca farmers"), where *"Coca"* is a proper noun denoting the coca plant. All errors we found in this category either contain proper nouns or foreign language material, which both typically lead to coverage issues in lexicon-based NLP applications (like e.g. parsers or in our case a rule-based morphological analyser).

Desired Analysis Missing In the second group, we give examples where Smor returns analyses, but the desired analysis of the gold standard is missing. In the case of the family name "Walton", again, a respective lexicon entry is missing. Instead, Smor -d only retuns the analysis *"Wal|Ton"* (= "whale|tone"), which is correct from a morphological point of view, but semantically highly unplausible. A lexicon entry for the proper noun would have prevented the word from being split, regardless of the word part frequencies. In contrast, we observed some cases where missing analyses blocked words from being split, that should have been split according to the gold standard annotation. An example is the adjective *"treffsichere"* (= "certain hit"), for which SMOR only returns the analysis *"Treff|sichere"* (= "meeting|certain") instead of the correct *"treffen|sichere"* (= "to hit|certain"). Due to the frequency of the compound as a whole which is higher than the geometric mean of the part frequencies *"Treff"* and *"sichere"*, the word is left unsplit. However, the frequency score of the correct option *"treffen|sichere"* would have led to a splitting. Finally, we also found faulty splittings due to missing lexicon entries. *"Nord|europa"* (= "north|Europe") is a typical example for this group. Strictly speaking, *"Nordeuropa"* should be split into *"Norden|Europa"* (= "the north|Europe") instead, as the short form *"Nord"* is only used in nautical, poetic or meteorological language and is not common in everyday German.

Lexicalisations The largest group of errors in this evaluation comes from compounds that are lexicalised in SMOR ***"-d"***, and whose internal analyses are thus not accessible for splitting. Note that the errors of this group do not occur in the SMOR splitting without usage of the "-d" flag (see Section 7.2.1 for more details on the ***"-d"*** flag in SMOR). In our present evaluation, both *"Zielgruppe"* (= "target group") and *"Zentralbank"* (= "central bank") occur as one lexeme in SMOR's lexicon and are thus not split.

Hierarchy Errors Finally, some errors occur which are attributed to the inability of SMOR to output hierarchical analyses. As all analyses are flat, they might erroneously indicate split points. We already gave the example *"Untersuchungshäftling"* (= "person being imprisoned on remand") on page 101 above. Another example of this category is *"Schrittmacher"* (= "pacemaker"), which is split into *"Schritt|Macher"* (= "step|maker"). However, the internal structure of the word blocks a splitting into these two parts from a linguistic point of view: ((Schritt$_N$ + machen$_V$)+ -er$_{N-suffix}$), which is also the reason why the compound is left as a whole in the gold standard annotation. The analysis by SMOR correctly indicates the parts of the compound, namely *"Schritt|machen|er"*. Splitting into the nominal suffix *"-er"* is explicitly blocked, and the fact that the first split point between *"Schritt"* and *"machen"* lies on a different hierarchical depth is not apparent. As both, *"Schritt"* and *"Macher"* occur in the corpus, the compound is erroneously split.

Concluding Remarks Obviously, the accuracy of our splitting approach depends on the coverage of the rule-based morphological analyser it relies on. The gold standard evaluations showed that our approach reached the highest accuracies across all standards and settings. Moreover, we have seen from the error analysis that the errors of our approach are two-fold: frequency-related errors are due to the corpus-based disambiguation strategy we use to rank different splitting options and SMOR-related errors are attributed to missing coverage of SMOR. However, the detailed error analysis shows that only very few words are completely missing in SMOR's lexicon (cf. *unknown to* SMOR in Table 8.15 above), whereas most of the errors come from completely lexicalised compounds (cf. *lexicalised in* SMOR *-d*). This is exactly what our linguistically informed splitting approach aims at: compounds should be split whenever there is linguistic and corpus-based evidence for it. We want to produce high-precision splittings and therefore accept a slight loss of recall due to over-lexicalisations.

8.7. Chapter Summary

In this chapter, we presented three different kinds of gold standards in detail: one based on translational correspondences, a linguistically motivated one and an external domain-specific gold standard. Then, we investigated the accuracies of different splitting approaches, based either on corpus frequencies, POS or the rule-based morphological analyser SMOR, with respect to these gold standards. We applied standard evaluation metrics (based on precision and recall) to compare the splitting accuracies of the different approaches. Across all of the three gold standards we used, we could clearly see that the splittings of our morphologically-aware approaches were the most accurate ones.

Finally, we performed a detailed error analysis for the three kinds of splitting approaches and reported numerous typical error examples to give an insight into how the approaches work, aside from the accuracy numbers alone. The detailed analysis revealed two kinds of errors our morphology driven splitting approach suffers from: i) limited coverage of the rule-based morphology and ii) frequency-related errors due to the disambiguation procedure. In terms of quantity, we found only very few errors related to the coverage of the rule-based morphology. With regard to the high splitting precision of our approaches compared to the previous approaches, this slight loss of recall is acceptable.

In the next chapter, we will investigate the performance of the different splitting approaches in end-to-end SMT systems.

9. SMT Evaluation: German to English

In previous chapters, we introduced our morphologically-aware compound splitting approach (Chapter 7) and compared it to two less informed splitting approaches with respect to different gold standards (Chapter 8). In this chapter, we integrate morphological compound splitting into German to English SMT and compare it to a baseline system without compound splitting and to the two linguistically less informed approaches introduced in Chapter 6 and show significant improvements.

Motivation Recall from Section 5.1.2 that the intuitive idea of using compound splitting in statistical machine translation systems builds on the assumption that making source and target language more similar on a morphological level will help word alignment and thereby also translation quality. By splitting compounds in the German source language which do not exist in English, we make the two languages more similar in terms of morphological granularity. Moreover, compounds that have not occured in the training data and are thus not translatable by conventional SMT systems, can often be translated part-by-part, provided that they have been split correctly and their parts have occured in the training data.

In the following, we will show that these assumptions really hold: we integrate compound splitting into end-to-end SMT systems and find overall improvement in automatically measurable translation quality compared to a simple baseline system without compound splitting. Moreover, we also show that our morphologically-aware splitting, using a rule-based morphological analyser, significantly outperforms the re-implementations of linguistically less informed compound splitting approaches.

Previous Work Splitting compounds in German to English SMT has become common practise in recent years. See Chapter 10 for an extensive review of related work. As far as we know, only Popović et al. (2006) compared the influence of frequency-based and morphological compound splitting procedures in end-to-end SMT. However, Popović et al. (2006) found only slightly improved performance for the morphological compound splitting. In our experiments, we found statistically significant improvements of translation performance. In contrast to Popović et al. (2006), who used linguistic context for disambiguation, we use a frequency-based disambiguation of multiple, linguistically motivated, splitting options.

In Fritzinger and Fraser (2010), we repored on similar SMT results. These are comparable to the results we present in this chapter. However, they were calculated for an earlier version of the compound splitting procedure.

Structure This chapter is structured as follows: in Section 9.1, we describe the parameters of the underlying SMT architecture we use and give details on the different splitting experiments we implemented. We report SMT results for all systems in Section 9.2, where we give BLEU scores, unknown word counts and some translation examples. The findings of this chapter are summarised in Section 9.3.

9.1. Experimental Setting

In the following, we introduce all parameters of our SMT experiments. The section consists of two parts: technical details of the SMT systems can be found in Section 9.1.1 and an overview of the splitting approaches under investigation is given in Section 9.1.2.

9.1.1. Translation system

In all experiments we present in this chapter, compounds are split prior to translation model training. In order to ensure that all measured effects on translation quality are attributable to this pre-processing, we build identical translation systems for each compound splitting approach under investigation. This means the systems differ only in the way the German source language data was split (or not).

Data We use data from the shared task of the EACL 2009 workshop on statistical machine translation.[47] The training data consists of ∼1.5 million parallel sentences (∼40 million words). It is composed of the proceedings of the European parliament debates (cf. EUROPARL corpus, version 4, Koehn, 2005) and some news texts. We removed all sentences longer than 40 words from the training corpus. Moreover, we normalised orthographical variants of German words that were due to the German writing reform, see Fraser (2009).

We use 1,025 sentences for tuning and 1,026 sentences for testing. All data was lower-cased and tokenised, using the shared task tokeniser. Then, we split compounds in the German sections of the bilingual training, tuning and testing data using the different compound splitting approaches.

Language Model Based on the English monolingual training data of the shared task (containing roughly 227 million words), we trained a 5-gram language model using the SRILM toolkit (Stolcke, 2002) with Kneeser-Ney smoothing. We then use KenLM (Heafield, 2011) for faster processing. As compound splitting only concerns the German source language side, this English language model can be used for all experiments as is.

Translation Model For word alignment, we used the multithreaded GIZA++ toolkit (Och and Ney (2003), Gao and Vogel (2008)). We use Moses, a toolkit to build phrase-based statistical machine translation systems[48] (Koehn et al., 2007) to train a translation model and for decoding. For each splitting approach under investigation we built a separate system. We did so by closely following the instructions of the shared task[47], using only default parameters.

Tuning For tuning of feature weights, we ran Minimum Error Rate Training (Och, 2003) with Batch-Mira (Cherry and Foster, 2012) and *"-safe-hope"* until convergence (or maximally 25 runs), optimising BLEU scores (Papineni et al., 2002). We ran tuning individually for each system.

Testing After decoding, the output text was automatically recapitalised and detokenised, using the tools provided by the shared task. For translation quality evaluation,

[47] `http://www.statmt.org/wmt09/translation-task.html`

[48] The Moses toolkit can be obtained from `http://www.statmt.org/moses/`; we used version 1.0.

name	**subword restrictions**	**filler letters**	**delet. letters**	**stop words**	**other**	**to be split**	**details in Section**
basic freq	min. 3 character	limited	no	no	no	all	6.2.1
extended freq	min. 4 character	extended	yes	yes	f $\geq$ 3	all	6.2.2
Pos	corpus cleaning	extend.	yes	yes	pos-constraints	nouns	6.3
Smor	Smor	n.a.	n.a.	no	deep analysis	all	7.1
Smor -d	Smor	n.a.	n.a.	no	flat analysis	all	7.2.1
Smor -d NN	Smor	n.a.	n.a.	no	flat analysis	nouns	7.2.1

Table 9.1.: Summary of compound splitting approaches which we compared and evaluated in end-to-end Smt.

we calculated Bleu (Papineni et al., 2002) and Meteor scores (Lavie and Agarwal, 2007). We used version 11b of Bleu and for Meteor we used 1.4, *"exact stem synonym paraphrase"*, with weights *'1.0 0.6 0.8 0.6'*. See Section 4.2.3 above for more details on how exactly these two metrics approximate translation quality. For the Bleu metrics, we performed significance testing using pair-wise bootstrap resampling with sample size 1,000 and a p-value of 0.05.[49]

9.1.2. Experiments

In order to investigate the effect of compound splitting on Smt performance, we implemented one system without any compound processing and compared its outcome to several translation systems for which compounds were split prior to training.

Raw Baseline *Raw* denotes a very simple contrastive system which we build by closely following the instructions of the shared task for the construction of a baseline system.[50] For each step, we used only default parameters and we did not perform any kind of pre-processing or post-processing on any of the data sets.

Compound Splitting We give an overview of the splitting approaches for which we built Smt systems in Table 9.1. Note that this Table is a reproduction from Table 8.1. In Table 9.1, we distinguish three main types of different approaches: i) frequency-based, ii) Pos-based and iii) Smor-based (= our approach). All approaches have been introduced in great detail in Chapters 6 and 7.

[49] The code can be obtained from `http://www.ark.cs.cmu.edu/MT`

[50] These can be obtained from `http://www.statmt.org/wmt09/baseline.html`

	mert.log	**Bleu**	**Bleu ci**	**RTS**	**Meteor**
raw	20.66	18.99	21.40	1.0021	29.07
basic freq.	20.66	18.27	20.70	1.0027	29.53
extended freq.	21.31	19.19	21.86	1.0004	30.15
Pos	21.47	**19.27**	21.89	1.0062	30.16
Smor	21.73	**<u>19.42</u>**	22.11	1.0003	30.32
Smor -d	21.72	**<u>19.82</u>**	22.51	0.9975	30.40
Smor -d NN	21.57	**<u>19.60</u>**	22.21	0.9959	30.17

Table 9.2.: End-to-end Smt results for different compound splitting approaches. Statistically significant differences were calculated for the case-sensitive Bleu scores. **Bold**-faced scores indicate statistical significance wrt. the *raw* baseline, <u>underlined</u> scores are also significant wrt. the *extended freq.* scores.

9.2. Results

In this section, we give end-to-end Smt results for each of the compound splitting approaches under investigation. We first report measurable effects using standard automatic MT evaluation metrics in Section 9.2.1. Then, we give details about the impact of compound splitting on vocabulary size reduction in Section 9.2.2. The section concludes with some handpicked examples of erroneously translated segments originated from erroneous splittings in Section 9.2.3.

9.2.1. Translation Quality

As mentioned earlier, we evaluated all Smt experiments with two standard MT evaluation metrics: Bleu (Papineni et al., 2002), version 11b, and Meteor (Lavie and Agarwal, 2007), version 1.4. More details on these metrics can be found in Section 4.2.3. The results for all splitting approaches are given in Table 9.2.

There, *mert.log* denotes the (case-insensitive) Bleu score of the final minimum error rate training run of the tuning phase. Bleu denotes the case-sensitive Bleu score of the respective MT output with respect to the reference test set. Bleu *ci* is the **c**ase-**i**nsensitive version thereof, i.e. it shows improvements that are independent of the recasing model, moreover. *RTS* denotes the length penalty. Finally, Meteor gives the actual Meteor score, calculated using the default parameter settings of Meteor. For Bleu, we even calculated significance with respect to the *raw* baseline and the extended frequency-based approach respectively.

First of all, the numbers in Table 9.2 show that compound splitting in general has an overall positive effect on the automatically measurable translation quality. Recall that all systems were tuned to optimise BLEU scores. It thus happens that we have systems that show quite large improvements in BLEU, whereas there is hardly an improvement in METEOR (e.g. *extended freq* vs. POS) and vice versa (e.g. *raw* vs. *basic frequency*).

Taking a closer look, we find that the BLEU scores of the *basic frequency* based approach drop slightly, when compared to the *raw* baseline, whereas in contrast, METEOR scores slightly increase. The *extended frequency-based* approach performs better, both in terms of BLEU scores (+0.2 points) and more than one point for METEOR (+1.08 points). Interestingly, the POS-*based* approach scores significantly better than the *raw* baseline in terms of BLEU scores (+0.28) but only minimally better than the *extended frequency-based* approach, both in terms of BLEU and METEOR.

Moreover, it can be seen from Table 9.2 that the three SMOR-*based* approaches score highest in BLEU (with all scores being significantly better than the *raw* baseline and the extended frequency-based approach) and METEOR. Compared to the *raw* baseline, the absolute improvement of the best approach, SMOR *-d*, is +0.83 BLEU and +1.33 METEOR points. Generally speaking, we can say that the splitting approaches leading to the best translation quality in our experiments are the same that scored highest in our gold standard evaluations (as reported in Chapter 8 above).[51]

9.2.2. Vocabulary Reduction / Unknown Words

Recall from Section 5.1.2 above that traditional SMT approaches suffer from data sparsity when translating from a compounding into a non-compounding language. Due to the productivity of compounding languages, many source words may not have occurred in the parallel training data (or not often enough) and thus could not be learned adequately in the translation model. Compound splitting is known to reduce the vocabulary: Berton et al. (1996) showed a vocabulary reduction of 24% and noted that the larger a dataset, the more compound splitting helps to reduce its vocabulary. In SMT, a smaller vocabulary contributes to improved translation quality. However, the vocabulary should not be reduced too much either: ideally, compound splitting should lead to a balanced number of source and target language words in order to get optimal translation quality.

[51] This observation stands in contrast to findings of (Koehn and Knight, 2003), whose experiments showed that the most accurate splitting wrt. their gold standard did not score highest in BLEU.

system	**parallel data**			**test data**		
	tokens	**types**	**singletons**	**tokens**	**types**	**unknown**
raw	39,411,521	345,361	173,455	23,931	6,700	921
basic freq.	44,373,985	87,405	25,328	27,483	6,222	719
extended freq.	41,928,277	131,183	49,247	25,704	6,327	612
Pos	41,118,885	185,645	82,148	25,188	6,385	652
Smor	41,937,423	140,518	53,129	25,641	6,283	590
Smor -d	41,397,726	148,045	55,430	25,338	6,362	586
Smor -d NN	40,766,342	178,046	73,818	24,962	6,488	647

Table 9.3.: Vocabulary sizes for the German source language files for training and testing.

In Table 9.3, we give the vocabulary sizes of the German section of the parallel data and the German test set after applying each of the splitting approaches. In addition, we calculated the number of words occurring only once in the training data (*singletons*), for which a correct translation cannot be guaranteed due to their low frequency. For the test set, we also calculated the number of word types that have not occurred in the parallel data. These *unknown* words are left untranslated by the systems. They are passed through the decoder "as they are". Note however, that not all of these are unknown words in the sense that they have a negative influence on translation coverage. Often, these are proper nouns for which translation is not necessary anyway.

Taking a closer look, we can see from Table 9.3 that the *basic frequency-based* splitting drastically reduced the vocabulary of the training data from 345,361 word types to less than a third of it, namely 87,405. In contrast, the number of unknown words only moderately drops from 921 in the original test set to 719 in the split version. It is quite obvious that this approach tends to "over-split" the German source language, i.e. it makes it more fine-grained than the English counter part. Even though phrase-based Smt can easily recover from such over-splittings by memorising erroneous splittings as phrases, we have seen from the translation score results in Table 9.2 that the frequency-based splittings improved translation quality less than the more accurate linguistically motivated splittings of the Smor-based approaches.

Another interesting observation from Table 9.3 is the vocabulary size comparison of the two morphologically-aware splitting approaches: it shows that Smor *-d* leads to less unknown words in the test data, even though its training vocabulary was larger. These findings are in line with the translation quality results we discussed above.

Reference	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **warenhäusern** gar nicht mehr auf die preise schaut .
	we found that most shoppers in the three **outlets** visited were not really interested in prices .
extended freq.	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **waren häusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **were houses** no longer on prices .
Smor -d	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **warenhäusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **department stores** no longer on prices .

Table 9.4.: Translations of sentence Nr. 31 of the wmt 2009 test set for the extended frequency-based splitting approach and the morhpology-driven approach (Smor -d). The example shows that erroneously splitting the lexicalised compound *"Warenhäuser"* ("warehouses") may lead to a translation error.

9.2.3. Translation Examples

In this section, we show negative translation examples of the *extended frequency*-based system and the Smor *-d* system for two sentences of the test set, one containing a lexicalised compound that should not necessarily have been split, and one containing a compositional compound that should definitely be split, but where the correct splitting nevertheless leads to a wrong translation.

Lexicalised Compounds In Table 9.4, we give the translations for sentence Nr. 31 of the wmt 2009 test set, where the German input differs in the splitting of *"Warenhäusern"* (= "warehouses"). As in English, the word is lexicalised in contemporary German and should thus be left unsplit, as happened in the case of the Smor *-d* preprocessing. This yields a correct (even though not exactly reference-matching) translation into "department stores". However, *"Warenhäusern"* has been erroneously split into *"waren"* (= "were") and *"Häusern"* (= "houses") by the *extended frequency-based* approach and this leads to the erroneous translation "were houses". While a split of *"Warenhäusern"* into *"Ware|Häusern"* (= "ware|houses") would be acceptable from a linguistic point of view, the splitting into the (more frequent) verb *"waren"* (= "were") is not. This example motivates the use of a high precision splitting approach like Smor *-d* which contains many lexicalised compounds (as we have already illustrated qualitatively with the error anal-

Reference	auch der **seitenhalt** hat sich gegenüber dem 206 wesentlich verbessert .
	also , the **lateral seat support** has improved markedly in comparison with the 206 model .
extended freq	auch der **seit halt** hat sich gegen über dem 206 wesentlich verbessert . *also the **since support** has against over the 206 considerably improved .*
	the **since cohesion** has 206 considerably .
Smor -d	auch der **seite halt** hat sich gegenüber dem 206 wesentlich verbessert . *also the **side support** has in comparsion with the 206 considerably improved.*
	the **hand** has **confined** itself to the 206 considerably .

Table 9.5.: Translations of sentence Nr. 328 of the wmt 2009 test set for the extended frequency-based splitting approach and the morhpology-driven approach (Smor -d). The example shows that a correct splitting of *"Seitenhalt"* (= "lateral support") into *"Seite|Halt"* (= "side support") does not guarantee a correct translation. Glosses are given in *gray*.

ysis in Section 8.6.3). We give the translations of all other splitting approaches under investigation in Table C.1 of Appendix C.

Correct Splitting ≠ Correct Translation The second set of translation examples concerns sentence Nr. 328 of the wmt 2009 test set and is given in Table 9.5. Note that we give *glosses* in this table, as the input and reference translations have different syntactic structures. Again, we compare the split input and translation output of the *extended frequency-based* splitting approach and that of Smor *-d*. In Table C.2 of Appendix C, we give the translations of the other approaches, too. This time, the sentence contains a compositional compound, *"Seitenhalt"* (= "lateral support") that should be split into *"Seite"* (= "side") and *"Halt"* (= "support"). As can be seen, the compound is correctly split by Smor *-d*, while the *extended frequency-based* system erroneously splits into *"seit|Halt"* (= "since|support") instead. Despite the correct splitting, the system using the Smor *-d* splitting translated *"hand confined"* which has nothing to do with the meaning of "lateral support". A closer look reveals that *"Seite"* was aligned with "hand" (probably originating from the expression "on the other hand" - *"auf der anderen Seite"*) and *"Halt"* was aligned to "confined". However, the output of the *extended frequency-based* ("since cohesion") does not reflect the correct meaning of the German compound either. Comparing the phrase table entries for *"Halt"* of the *exended frequency-based* experiment and the Smor *-d* experiment, we found that "cohesion" and "confined" both occured in the two phrase tables, but "cohesion" got higher probabilities for *"Halt"* in the *extended* *frequency-based* experiment, while "confined" scored higher in the Smor *-d* experiment.

This example shows that a correct splitting does not neccessarily lead to a correct translation. Nevertheless, correct splittings are still good prerequisites for better translations, as the overall improved translation quality scores showed. Recall also from Section 4 that translations obtained through Smt depend on the interaction of numerous different components, which makes translations sometimes difficult to understand.

9.3. Chapter Summary

In this chapter, we gave detailed technical descriptions of the German to English translation systems into which we integrated different compound splitting approaches in order to measure their impact on translation quality. The results in terms of automatically measurable translation metrics showed that our morphologically aware compund splitting approach clearly outperforms previous, linguistically less informed approaches. This is in line with conclusions drawn from the gold standard evaluations in the previous Chapter 8, where we showed that the Smor-based approaches produce the most accurate splittings on clean data. Having a closer look at the vocabulary sizes of the training and test data, we found evidence that the frequency-based approaches often "over-split" words and compounds. This reverses the imbalance of morphological granularity between compounding and non-compounding languages and leads to suboptimal translation results.

In the next part of the thesis, we will show that compound processing significantly improves translation quality in the opposite translation direction as well, from English to German. This direction is more challenging as compounds do not only have to be split accurately, but also have to be re-synthesized appropriately.

10. Related Work

In the past, there have been numerous attempts to integrate compound splitting in SMT, all of which achieved improved translation performance due to compound processing. In this section, we sketch previous work on compound splitting across different splitting procedures (Section 10.1), disambiguation strategies (Section 10.2), segmentation levels (Section 10.3) and along the way give examples for compound splitting applications beyond SMT. In Section 10.4, we summarise related approaches on word packing.

10.1. Splitting Procedures

Previous splitting approaches differ with respect to the amount of linguistic knowledge they incorporate. High level linguistic approaches usually lead to precise splittings on the one hand, but on the other hand they make the procedure language-dependent and thus less transferable to other languages. We present an overview of related work ranging from linguistically less informed approaches (e.g. frequency-based splittings) to more informed approaches incorporating hand-crafted linguistic resources (e.g. dictionaries).

Frequency-based Splittings In Chapter 6, we already presented the frequency-based splitting approach of Koehn and Knight (2003) in detail. Frequency-based splitting approaches are simple and very effective. They perform compound splitting by using substring corpus frequencies and a minimal amount linguistic knowledge (allow filler letters "*(e)s*" for German). Koehn and Knight (2003) also reported on additional experiments with POS restrictions on the compound parts and translational correspondences. However, these could not improve over the SMT performance of the frequency-only splitting. The frequency-based approach of Koehn and Knight (2003) has been frequently re-implemented, with variations regarding filler letters, minimal part sizes and frequencies e.g. by Popović and Ney (2004), Yang and Kirchhoff (2006), Fraser (2009) and Durgar El-Kahlout and Yvon (2010).

Stymne (2008) extensively investigates different parameter settings of the approach of Koehn and Knight (2003) and, most importantly, enriches the splitting procedure with POS-based constraints: i) allow only splits into content words and ii) the POS of the compound head (= the rightmost word part) must match the POS of the whole compound. Moreover, she introduces a special POS markup scheme for compounds (where the POS of former compound modifiers and heads differ) and she also makes use of factored translation models (Koehn and Hoang, 2007) and an additional higher gram POS language model. In contrast to previous experiments by Koehn and Knight (2003), she finds moderate translation quality improvements due to the POS constraints. Weller and Heid (2012) describe a similar splitting procedure using substring frequencies and POS constraints for the task of terminology alignment and acquisition.[52] Alfonseca et al. (2008b) train a support vector machine for compound splitting whose features rely on the frequency-based splitting score of (Koehn and Knight, 2003) but also other frequency scores such as compound probability and mutual information.

Other frequency-based splitting approaches make use of letter-based n-grams (Larson et al., 2000; Sereewattana, 2003) or statistical correlation of compound parts (Zhang et al., 2000) to identify suitable compound splits. More recently, Macherey et al. (2011) implemented an unsupervised frequency-based approach to compound splitting where morphological operations, compound part boundaries and a stop-list of words not to be split (e.g. proper nouns) are learned from phrase tables.

Alignment-based Approaches In order to find the optimal segmentation level for word alignment (and thus machine translation), some approaches to compound splitting perform two-pass word alignments: in the first pass, word alignment is run on a split representation of the morphologically rich language. After that, all split words remaining un-aligned (though having no corresponding counterpart in the other language) are either deleted or merged back to the compounds before the second word alignment pass, see for example DeNeefe et al. (2008) for experiments on Arabic, Bai et al. (2008) for Chinese, or Koehn and Knight (2003) for German; Similarly, Lee (2004) uses the POS of the words a former Arabic compound part is aligned to as an indicator for compound splitting. If the POS differs from the POS of the compound, the part is split from the word stem.[53]

[52] Note that we used re-implementations of the approaches of Koehn and Knight (2003) and Weller and Heid (2012) in order to compare their translation performances to our own splitting approach.

[53] Note that Arabic word formation works through a combination of word stems with articles, prepositions, personal pronouns and the conjunction corresponding to “and” in English. This approach

Brown (2002) describes an alignment-based approach to compound splitting based on cognates.[54] This approach is designed for compounds from specialised domains (e.g. medical texts), where many out-of-vocabulary words have a common Latin or Greek origin and their translations are thus orthographically very similar to the source word. Split points are identified by merging adjacent words in the non-compounding language and measuring their similarity to the corresponding word in the compounding language.

The compound splitting approach by Popović et al. (2006) does not directly make use of word alignment for compound splitting, but instead, compounds are split only for word alignment. Before translation model training, the alignments to positions of the former compound parts are re-aligned to the position of the compound.

Unsupervised Approaches Unsupervised approaches to learning the morphology of a language based on unannoted text are a suitable alternative to hand-crafted rule-based morphological analysers, which are often unavailable. The main advantage of unsupervised approaches is their adaptation potential: provided the availability of a large amount of unannotated data, hardly any language-specific knowledge is required to learn the morphology of any language or even specialised domain. This distinguishes them from hand-crafted rule-based resources which require (specialised) lexicons for each language and domain. However, as unsupervised approaches often do not incorporate linguistic knowledge, their segmentations might consist of non-words and non-morphemes, which in turn may distort translation performance (depending on the language pair and translation direction).

In the past, the Morfessor toolkit (Creutz and Lagus, 2005) has been used repeatedly for compound splitting in SMT, e.g. (Virpioja et al., 2007), (Fishel and Kirik, 2010), or (Luong et al., 2010). Morfessor's underlying algorithm consists of two main steps: First, a lexicon of morphemes is learned from unannotated text. Then, all morphemes are categorized into prefixes, stems and suffixes. Virpioja et al. (2007) used Morfessor for morpheme-based translation of Scandinavian language pairs (DK, SV, FI) and, despite no measurable translation improvements, the unsupervised approach had a positive effect on out-of-vocabulary rates and enabled generalisation over words and word parts. More recently, Fishel and Kirik (2010) reported results of translation from English into

could hardly be applied to compounds of Germanic languages.

[54] Cognates are words which share the same etymological background. An example taken from Brown (2002) is: *"Abdominalangiographie"* - "abdominal angiography".

Estonian and vice versa, where the unsupervised approach outperformed rule-based segmentation in end-to-end SMT performance. Luong et al. (2010) use MORFESSOR to learn Finnish morphology and apply it in a combined morpheme- and word-based translation model for the translation pair English to Finnish.[55]

Demberg (2007) starts from another existing unsupervised approach (REPORTS, Keshava and Pitler, 2006), which is well-established for English, and adapts it to the needs of morphologically richer languages, for example by imposing morphotactic constraints (realised via a bigram language model on morphemes). An evaluation within the framework of grapheme-to-phoneme conversion shows that the extended REPORTS outperforms MORFESSOR and is competitive (in terms of accuracy) with rule-based approaches like SMOR.

While the unsupervised approaches mentioned above aim at learning the morphology of a language, Bai et al. (2008) implemented a purely word-alignment driven approach for Chinese, where only splits into units that have an own corresponding word in the target language are feasible. Similarly, Chung and Gildea (2009) present different unsupervised tokenisation models for translation from Chinese and Korean to English.

(Semi-) Supervised Approaches A semi-supervised approach to Arabic word segmentation has been described by Lee et al. (2003). They use hand-crafted tables of prefixes and suffixes and a small manually segmented corpus to derive possible split points, which are then disambiguated using a trigram language model on morpheme level. New stems are acquired by iteratively applying the approach on unannotated text. Similarly, El Isbihani et al. (2006) implemented two small hand-crafted finite state automata (FSAs) covering Arabic prefixes and suffixes (and concatenations thereof), but without stem lexicon. Optimal word segmentations are learned by iteratively applying the FSAs to an unannotated corpus: parts are split from the stem only if they are found in the corpus.

Baldwin and Tanaka (2004) use syntactic templates filled with word translations from bilingual dictionaries to train a support vector machine (SVM) to translate compositional noun compounds within the English-Japanese language pair. In contrast, Alfonseca et al. (2008b) train a SVM with less linguistic knowledge, based on different corpus-based frequency and probability scores for German compounds and their parts.

[55] Some details of this morpheme-word representation are given in Section 10.3 below.

Rule-based Morphological Analysers Nießen and Ney (2000) were the first to use a rule-based morphological analyser for compound splitting in statistical machine translation. They applied the commercial morphological analyser Gertwol (Koskenniemi and Haapalainen, 1996) to split compounds in German to English SMT. More recently, Hardmeier et al. (2010) used Gertwol in SMT not only for compound splitting but also to access word lemmas and use them as alternative translation possibilities in a lattice-based SMT translation setting. In (Fritzinger and Fraser, 2010), we describe our own work on compound splitting in German to English SMT using the rule-based morphological analyser Smor (Schmid et al., 2004);

For Arabic to English SMT, Habash and Sadat (2006) applied Bama, the Buckwalter Morphological Analyser (Buckwalter, 2002), to split words into stems, clitics, particles and affixes. They found that such morphological preprocessing helps where only small training data sets are available, but given huge training data, the preprocessing can harm translation quality performance. In the same year, Zollmann et al. (2006) used Bama in a speech translation task.

Oflazer and Durgar El-Kahlout (2007) use a rule-based analyser for Turkish (Oflazer, 1994) to segment words into morpheme for translation from English into Turkish. The same analyser was later applied by Bisazza and Federico (2009) to split words into morphemes for the opposite translation direction in a spoken language translation task.

Note that rule-based morphological analysers have also been used for compound splitting in other NLP applications, e.g. by Schiller (2005) or Marek (2006), who implemented weighted finite-state automata; Braschler and Ripplinger (2004) or Moulinier et al. (2001) for information retrieval, Spies (1995) for automatic speech recognition, or Baroni et al. (2002) for a word prediction system.

Other Hand-crafted Resources So far, we reported on different approaches to compound splitting ranging from corpus-based metrics to the use of rule-based morphological analysers. More easily, compound splittings can also be derived from hand-crafted linguistic resources which are annotated accordingly, see e.g. Carlberger and Kann (1999), Monz and de Rijke (2001), and Garera and Yarowsky (2008) who used lexicon-lookups or Berton et al. (1996) and Goldwater and McClosky (2005) who used treebanks for compound splitting. While such lookup-based approaches lead to high precision splittings, their recall is limited through the resource's lexical coverage. Another drawback ist the lack of ability of such high-quality linguistically annotated resources.

10.2. Disambiguation Strategies

In the previous section, we presented different splitting procedures, i.e. strategies to identify possible split points of compounds. However, since many compounds can have more than one possible splitting option, a disambiguation routine is required to find the one best splitting to use in machine translation. Here, we present different disambiguation strategies that have been used for compound splitting in the past. The section concludes with a paragraph on previous work in lattice-based machine translation, where no disambiguation is required prior to decoding, but instead, all different options are bundled in a lattice and handed over to the decoder, which in turn then identifies the most suitable option at runtime.

Frequency-based Disambiguation Previous approaches widely used *frequency-based disambiguation* of different compound splitting options. It is based on the intuitive idea to use corpus-derived frequencies of potential compound word parts and also of the compound as a whole to determine whether or not the compound should be split and how it should be split. In the classical approach, as first described by Koehn and Knight (2003), the geometric mean of compound part frequencies is used to score different splitting options. This disambiguation has then also been used by others, for example in our own work Fritzinger and Fraser (2010) or by Weller and Heid (2012); On the other hand, Stymne (2008) extensively compared different splitting strategies and types of numeric scores and found that the harmonic mean score turned out to perform best, as it is less conservative (= leads to more splits) compared to the geometric mean score. Popović and Ney (2004) used the harmonic mean score to disambiguate splits of Serbian words, as the geometric mean score (in absence of a minimal part size constraint) often leads to undesired splits into single characters. Similar to our work, Schiller (2005) disambiguated multiple analyses from a finite-state morphology. However, she modified the frequencies of the compound parts in order to prioritise compounds with the least number of parts.

Classifier-based Disambiguation Habash and Rambow (2005) describe an SVM-based disambiguation approach for Arabic trained on a set of 10 features, including e.g. POS, definiteness, gender, and number; Similarly, Alfonseca et al. (2008b) train an SVM to disambiguate German compound splitting options. However, in contrast to Habash and

Rambow (2005) who used linguistic features to disambiguate Arabic compounds, Alfonseca et al. (2008b) use frequency-based features like mutual information or log-likelihood scores. Oflazer and Durgar El-Kahlout (2007) use the supervised classifier of Yuret and Türe (2006) to disambiguate Turkish morphological analyses.

Context-sensitive Disambiguation Nießen and Ney (2000) use a rule-based morphological analyser to find compound split points and disambiguate them using a parser. This kind of context-sensitive disambiguation is useful to disambiguate for example the POS of the compound head, but due to many ambiguities even within one and the same POS, this method might not be sufficient to find the correct analysis. In our own compound splitting approach, we also make use of a parser for pre-disambiguation of possible split options. However, in our case, we use the parser only to prevent proper nouns from being split.[56]

No Disambiguation Instead of selecting a single best splitting option for each compound prior to translation, lattice-based approaches allow to pass a set of different splitting alternatives to the decoder an thereby postpone the final decision to a later stage in the translation process where the actual context might help to find a decision.[57] Xu et al. (2005) describe a lattice-based approach to English-Chinese translation which allows for character-level translation. This approach is extended by Dyer et al. (2008) who, in addition, used a sophisticated reordering model and applied it to Chinese to English and Arabic to English phrase-based and syntax-based SMT. In (Dyer, 2009), he uses a maximum entropy model to get concurring compound splitting options for German, Hungarian and Turkish, and translates from all of these languages into English, again, using lattices. Other previous work on lattice-based compound splitting approaches in SMT include (DeNeefe et al., 2008) for Arabic and (Hardmeier et al., 2010) for German, who, in addition to different segmentation levels also includes lemmatised versions of the compound parts in the lattices.

Similar to the lattice-based approaches, the back-off models described by Yang and Kirchhoff (2006) allow to postpone the selection of a single-best splitting option. The underlying idea of these back-off models is to use probability distributions on different

[56] Recall from Section 7.1.5 that some German proper nouns coincide with compositional compounds that should be split whenever they do not occur as a proper noun.

[57] In a sense, this is also a variant of context-sensitive disambiguation.

levels of specificity: in case the more specific representation (e.g. the translation of a compound as a whole) fails, the approach backs off to a more general representation (e.g. the lemma of the compound, or, at the next level, the parts of the compound). A similar approach for back-off lemmatisation of Estonian to English SMT is described in (Fishel and Kirik, 2010).

10.3. Segmentation Level

As the notion of a compound to be split differs across languages, the appropriate level of segmentation differs just as much. In this section, we give some details on segmentation levels of previous work with respect to different language or application requirements.

Language-dependent In German, compounds are a unit of two (or more) simple words that usually also occur independently as standalone words. The same holds for other compounding Germanic languages such as Swedish, Norwegian, Danish and Dutch. For all of these languages, compounds consist of simple (standalone) words, see e.g. Nießen and Ney (2000) for German, Stymne and Holmqvist (2008) for Swedish, and Monz and de Rijke (2001) for Dutch. Note that these languages also exhibit meaning-bearing derivational and inflectional affixes. However, as these affixes seldomly have standalone counterparts in English (which is also a Germanic language), they are usually not split from their stems,[58] e.g. *"machbar"* (= "doable"), which consists of the verb stem *"mach$_V$"* (= "do") and the derivational suffix *"-bar$_{Suff}$"* (= "-able").

In contrast, for a highly inflected language like Czech, compound splitting denotes splitting a word into its stem and inflectional affixes, the latter for which a less inflected target language like English might use function words see e.g. Goldwater and McClosky (2005). Arabic is another morphologically rich language with concatenative morphology. Here, compounds do not consist of multiple simple word stems, but instead, word stems are combined with articles, prepositions, pronouns and the conjunction corresponding to "and" in English. See Habash and Sadat (2006) and El Isbihani et al. (2006) for compound splitting in Arabic to English SMT and Naradowsky and Toutanova (2011) for a bilingual morpheme segmentation that improves word alignment quality between Arabic and English.

[58] There is one exception to this rule: as the Swedish definite article is realised through suffixation of the word, it might be useful to split it from the stem.

Finally, compound splitting is of particular interest for Finnish, as it is highly agglutinative. A Finnish compound word may consist of several simple words and suffixes that are each expressed by separate words in another language. Moreover it has a very rich verbal and nominal inflection (e.g. 15 grammatical cases which would often be expressed by prepositions in other languages), and also vowel harmony constraints. See Virpioja et al. (2007) for an unsupervised approach to Finnish compound splitting into morphemes, as preparation for morpheme-based translation of Finnish into Danish and Swedish (and vice versa).

Compound splitting has also been described for several other languages, including Turkish by Bisazza and Federico (2009), Hungarian (Novák, 2009), Chinese (Bai et al., 2008), Japanese (Tanaka and Baldwin, 2003), Korean (Yun et al., 1995) or Urdu (Habash and Metsky, 2008).

Different Concurrent Segmentation Levels Another group of previous approaches uses representations of two different, concurrent segmentation levels, i.e. word-based vs. morpheme-based. Oflazer and Durgar El-Kahlout (2007) describe morpheme-based English to Turkish machine translation for which they use a word-based language model to re-score the morpheme-based translations prior to phrase selection. More recently, Luong et al. (2010) took one step further and implemented a fully integrated word- and morpheme level aware translation model for English to Finnish SMT. It includes a *"word-boundary aware morpheme-level phrase extraction"* and makes use of a twin translation model to account for both segmentation levels.

Application-dependent Recall that the underlying idea of decompounding words prior to machine translation is to assimilate the levels of granularity of a compounding source language to a non-compounding target language (or vice versa). Thus, for machine translation, source and target language must be considered in order to find an optimal segmentation level, ideally leading to an enhanced number of 1-to-1 alignments. There have been some approaches in the past which explicitly make use of word alignment information to find optimal split points (e.g. DeNeefe et al. (2008) for experiments on Arabic, see section 10.1 for more)

However, compound splitting is also crucial for NLP applications beyond machine translation. In general, all applications that suffer from out-of-vocabulary words can benefit from compound splitting. Note also that compound splitting optimised for machine translation might not be optimised for an Information Retrieval (IR) task (and vice versa). For the latter, it is most beneficial to split a word into its stems and stripping inflectional affixes in order to enhance retrieval recall. In contrast, for SMT inflectional affixes should only be split from their stems when they either have a standalone counterpart in the target language. Previous work on compound splitting for IR tasks can be found in (Monz and de Rijke, 2001), (Braschler and Ripplinger, 2004), (Airio, 2006) and (Zhang et al., 2000).

Another example application is automatic speech recognition, where it is desirable to constraint compound parts to consist of a minimal number of phonemes, as very short units are easily misrecognised (Berton et al., 1996). Other compound splitting approaches in the field of automatic speech recognition have been described by Spies (1995), Adda-Decker et al. (2000), Larson et al. (2000). For speech generation, the correct segmentation is crucial for correct pronunciation, as the example of *"Staubecken"* illustrates: Staub|Ecken (= "dusty corners") is pronounced [ʃtaʊ**p**|ɛkən], while Stau|Becken (= "dammed reservoir") is pronounced [ʃtaʊ|**b**ɛkən].

10.4. Source Language Word Packing

Beyond the classical compound splitting approach to handle compounds for translations from a compounding to a non-compounding language, there have also been some approaches which implement compound processing differently, some of which we briefly mention here. Instead of splitting compounds in the compounding language in order to improve word alignment quality through more 1-to-1 alignments, one could also merge the words corresponding to compounds in the non-compounding language into bigger units. Popović et al. (2006) present an approach where consecutive English nouns and, moreover, all English words that were aligned to one German word are merged together before translation. Similarly, Ma et al. (2007) propose a method improve word alignment quality by iteratively merging English words that are aligned to one single Chinese word. This kind of merging applies not only to compounds but also to fixed expressions consisting of more than one word: in an early approach by Nießen and Ney (2000), 21 German multi-word phrases which represent distinct syntactic roles in the sentence (e.g. *"irgend etwas"* = "something") are merged into packages before translation into English.

Part III.

Compound Merging

Motivation The previous part of this thesis dealt with compound splitting and its integration into an German to English SMT system. In this third part of the thesis, we will focus on compound processing for the opposite translation direction, from English into German. This translation direction is challenging, as German is morphologically more complex than English in many respects. It is difficult to generate morphological phenomena of German, that are either not present or not marked in English. Examples include compounds and German nominal inflection. To account for compounds in translation from English to German, they first need to be split (as described in Part II) and lemmatised prior to SMT training. The SMT system translates from English into split and lemmatised German output format in which compounds then need to be re-merged and the whole text must be inflected to achieve fluent German output.

Contributions In this part of the thesis, we combine compound processing with inflection handling and integrate it into an end-to-end English to German SMT system. We use a rule-based morphological analyser to create an underspecified lemmatised representation of the German data, which allows for a maximal generalisation over former compound parts and separately occurring simple words. The same applies for inflectional variants thereof. After translation, word parts to be (re-)merged into compounds are identified using machine learning techniques. In contrast to previous work, we take the syntax of the English source language into account for the merging decision. Contrastive experiments show the feasibility of using information derived from the source language for the compound merging decision.

The comparison of SMT systems accounting for inflection handling with and without compound processing does not yield decisive results in terms of automatically measurable translation quality. However, in additional manual evaluation, we find a considerable improvement in terms of correctly translated compounds.

Structure The remainder of this part is structured as follows: we present our compound merging procedure in Chapter 11, including details on the underspecified representation, and on how compound merging is first predicted and then generated. In Chapter 12, we investigate the accuracy of different compound merging strategies, with and without considering the source language for the decision. We present end-to-end SMT experiments in Chapter 13, along with a detailed manual analysis of the compound translations. In Chapter 14, we review related works on compound merging and inflection prediction.

11. Compound Merging Procedure

The focus of this third part of the thesis is on SMT from English to German. While the issues of the opposite translation direction discussed in Part II remain – words that have not occurred in the parallel training data cannot be translated – they are aggravated by the fact that German inflectional morphology is more complex than English. For example, it requires the agreement of words within noun phrases (e.g. terms of *case* or *number*). Our compound merging procedure accounts for both the creation of compounds that have not occurred in the training data and context-dependent nominal inflection in the German output.

Motivation Due to their productivity, many German compounds might not have occurred in the parallel training data. In contrast, their simple component words are more likely to have occurred. This is the underlying idea of compound processing in SMT, to make the most out of the available information. Compound splitting makes the component words of a compound accessible to the translation model. After translation from English into split German, suitable simple words are to be (re-)merged into compounds. An illustrative example of how new compounds that have not occurred in the training data can be generated from translations of other compounds (or simple words) is given in Figure 11.1 (a similar version of which we already showed in Cap et al. (2014a), p.580). Assume that *"Obstkiste"* (= "fruit box") has not occurred in the parallel training data. By splitting compounds prior to training and merging compounds after translation, the translation of the modifier *"Obst"* (= "fruit") can be obtained from the translation of the (split) compound *"Obsthandel"* (= "fruit trading"), which has occurred in the data. The same applies for the head of the new compound, *"Kiste"* (= "box"), whose translation can be derived from the translation of the split compound *"Werkzeugkiste"* (= "tool box"). The parts of the new compounds could also have occurred in other compound positions or as simple words in the parallel data. For example, the translation of the modifier of the new *"Handelswerkzeug"* (= "trading tool") has been taken from the head

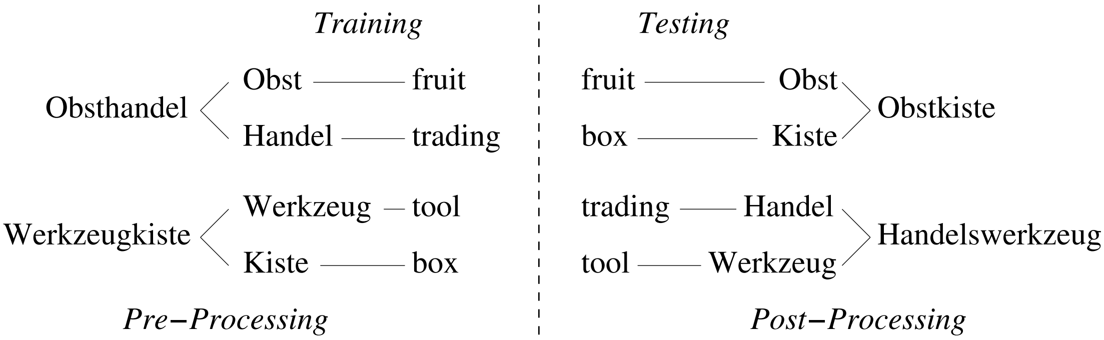


Figure 11.1.: Compound processing in English to German SMT allows the recombination of former compound parts into compounds unseen in the training data.

of *"Obsthandel"* (= "fruit trading") and vice versa for its head *"Werkzeug"*, which only occurred in the modifier position of *"Werkzeugkiste"* in the training data.[59] In order to allow words to occur in any compound position, all former compound parts must be reduced, i.e. morphological operations that were necessary for compounding must be reversed and later eventually re-generated, depending on the new position of the word.

However, compound processing in English to German SMT is not always as straightforward as the example in Figure 11.1 might suggest. For example, it is highly dependent on the quality of compound splitting: only a single correct prediction is required to merge the correct splitting of *"Niederschlag|Menge"* into *"Niederschlagsmenge"* ("amount of precipitation") but three for the linguistically not motivated erroneous split into *"nie|der|Schlag|Menge"* ("never|the|hit|amount").[60] Overall, the production of a compound which has not occurred in the parallel training data depends on:

1) precise compound splitting prior to training
2) the translation model choosing adequate translations of the compound parts
3) the decoder outputting the compound parts in the correct order
4) accurate identification of possible candidate words for compound merging
5) merging compounds by taking into account morphological operations
6) finding a suitable inflectional variant depending on the actual translation context

[59] From the training data as given in Figure 11.1, even the new compounds *"Kistenobst"* = "box fruit", *"Kistenhandel"* = "box trading", *"Obstwerkzeug"* = "fruit tool", *"Werkzeugobst"* = "tool fruit", *"Werkzeughandel"* = "tool trading", *"Handelsobst"* = "trading fruit" and *"Handelskiste"* = "trading box" could be generated.

[60] In contrast, they may not hurt translation quality in the other direction (from German to English), where phrase-based SMT may learn the split words as a phrase and thus recover from that error.

For compound splitting, we rely on our well-investigated morphological approach, which we presented in Chapter 7 above. However, in order to allow for inflection handling, we extended it to lemmatise not only former compound modifiers but also former compound heads.

Structure The remainder of this chapter deals with compound merging and is structured as follows: in Section 11.1, we introduce an underspecified representation which allows for free merging of former compound parts and simple words into compounds. Moreover it allows to adjust inflectional endings after translation. In Section 11.2, we describe Conditional Random Fields (CRFs), a machine learning method. We it use to predict possible candidate words for compound merging in Section 11.3 and to predict context-dependent inflectional endings in Section 11.4. After the predicting these features, we use the rule-based morphological analyser SMOR for the generation of full forms in Section 11.5.

11.1. Underspecified Representation

In Chapter 7, we already described our compound splitting procedure in great detail and investigated its effects on German to English SMT. For the opposite translation direction **into German**, we leave the general compound splitting process unchanged, i.e. we take possible split points from SMOR and disambiguate them using word part frequencies derived from a corpus. However, we adapt the output format to a *lemma-like* underspecified representation, consisting of the lemma of the word and relevant morphological features, similar to the *lemma-tag* representation of Nießen and Ney (2004). We use the underspecified representation of Weller (2009) which is compatible with SMOR. The reduced format is necessary in order to allow us to later freely recombine former compound parts and predict unseen inflectional variants. The underspecified representation consists of the lemma together with a set of morphological features, both of which we derive from SMOR. Generally speaking, we keep two types of features i) word-inherent features which are independent of the syntactical context (there is no benefit in predicting them) and ii) features which are distinguished (and visibly marked) in both source and target language. In the following, we describe the particularities of some word classes in more detail.

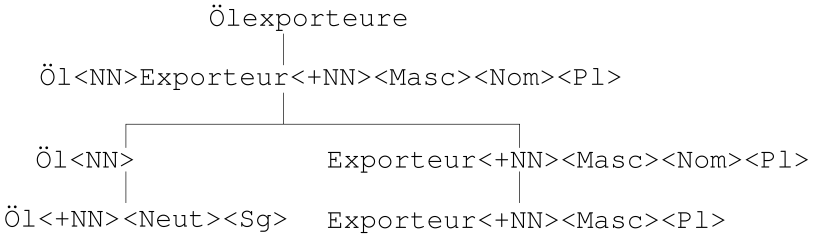

Ölexporteure
Öl<NN>Exporteur<+NN><Masc><Nom><Pl>
Öl<NN>
Exporteur<+NN><Masc><Nom><Pl>
Öl<+NN><Neut><Sg>
Exporteur<+NN><Masc><Pl>

Figure 11.2.: Transformation of compounds into the underspecified representation.[63]

Compounds Compound splitting and their transformation into the underspecified representation happens in one joint step.[61] An example is given for *"Ölexporteure"* (= "oil exporters") in Figure 11.2: it can be seen that SMOR outputs no morphological features for the former compound modifier, except for its part of speech (here: *"Öl<NN>"*). However, in order to allow for full generalisation over former compound parts and independent occurrences as simplex words, the underspecified representation of former modifiers must be identical to those of the same word when it occurred in isolation. We thus run the modifier separately through SMOR to receive the required additional features (here: *gender: <Fem>* = feminine and *number: <Sg>* = singular[62]) and at the same time to ensure full compatibility with simplex occurrences.

In contrast, the former compound head (here: *"Exporteure"*) automatically inherits all morphological features from the compound as a whole. From this set, only the context-dependent features must be deleted, e.g. *case* (here: *"<Nom>"*, nominative).

Portemanteaus Not only compounds, but also portemanteaus are split prior to training.[64] Portemanteaus are a fusion of a preposition and a definite article (thus not productive) and their *case* must agree with the *case* of the noun. For example, *"zum"* can be split into *"zu"* + *"dem"* = to+the$_{Dative}$. Portemanteaus introduce additional sparsity to the training data: imagine a noun occurred with its definite article in the training data, but not with the portemanteau required at testing time. Splitting portemanteaus allows a phrase-based SMT system to also access phrases covering nouns and their cor-

[61] Note that the transformation of compounds into the format, especially the handling of former modifiers, is original work presented here, whereas the details of the other word classes are reported here from previous work by Weller (2009).

[62] We chose the singular form as the default number for all former compound modifiers.

[63] We already showed this figure in Cap et al. (2014a), p.582.

[64] This description of portemanteaus is very similar to the one we gave in Cap et al. (2014b).

responding definite articles. We split portemanteaus based on a list[65] and their POS-tag *"APPRART"* which we get from BitPar parses of the German training data, then, in a second step, we lemmatise the articles, i.e. we remove all context-dependent features from their underspecified representation.

Thereby, we are able to abstract even more:[66] assume that the test set requires the translation into a dative phrase *"im neuen Auto"* (= "in the new car"), which has not occurred in the parallel training data. In contrast, *"in das neue Auto"* (= "into the new car"), which is the accusative realisation of the phrase, has occured. In the underspecified representation, the latter parts of these two phrases are identical: *"die<ART><def> neu<ADJ><Pos> Auto<+NN><Neut><Sg>"*. We can thus generalise from the accusative phrase seen in the training data to the dative phrase required from the testset, given that the preposition has occurred elsewhere in dative case, and then merge the preposition with the definite article, to make the sentence more fluent.

Nouns In German, nouns occur in different morphological forms depending on their *gender*, *number*, and *case*. For the underspecified representation, we keep *gender* and *number* and drop *case*. The gender of a German common noun is – except for very few cases – specific to the word and unique. Whenever SMOR returns multiple possible genders for a noun, e.g. *"Gehalt"* = "salary" (neuter) vs "content" (masculin), we use morphologically rich POS tags (containing gender information) which can be obtained from either a parser or a POS-tagger (we use BitPar, Schmid, 2004) to disambiguate the analyses.[67] Keeping the number feature leads to a loss in generalisation, but since there is a morphological distinction between singular and plural forms in English, too (in most cases marked with a *plural-"s"*), we expect the translation model to learn this distinction. Finally, the context-dependent *case* feature is deleted and will later be predicted to make the noun fit into its new syntactical context in the translation output.

Adjectives German adjectives must agree in *gender*, *number* and *case* with the noun they describe. Depending on the definiteness of the determiner, they occur in *strong* or *weak* inflection. The three different adjective singular nominative forms of the strong inflection for *"klein"* (= "small") in Table 11.1 illustrate this agreement contraint for

[65] of the 13 most common ones: *am, ans, aufs, beim, im, ins, übers, überm, vom, vor, zum, zur.*

[66] We already used a similar example *"im internationalen Rampenlicht"* in Fraser et al. (2012).

[67] Recall from Section 7.1.5 above that we also use POS tags obtaine from BitPar to distinguish between common nouns and proper nouns (page 84).

gender	**German**			**English**
masculin	ein	klein**er**	Knopf	*a small button*
feminine	eine	klein**e**	Maus	*a small mouse*
neuter	ein	klein**es**	Boot	*a small boat*

Table 11.1.: German adjectives must agree with their nouns in *gender*, *number* and *case*.

three nouns of different genders. In plural, the adjective form is no longer dependent on the *gender* of the noun, it is *“kleine”* for all. Beyond these, and the *case* variations, German adjective forms are also conform with the form of the article: in combination with indefinite articles, they occur in *strong* inflection (as can be seen from the *“klein”* example), whereas with definite articles they occur in *weak* inflection, which is one form independent of the noun gender, thus *“kleine”* for *“Knopf”*, *“Maus”* and *“Boot”*, respectively. As all of the features mentioned above are context-dependent, they are deleted for the underspecified representation, and will later be re-predicted in the translation output.

In contrast, we keep the distinction between positive (*“klein”* = “small”), comparative (*“kleiner”* = “smaller”) and superlative (*“kleinsten”* = “smallest”) form because this feature is morphologically marked in English as well.

Verbs The version of the re-inflection system we use does not allow for verbal re-inflection yet. We thus leave verbs in their full form (instead of their lemma) together with their part of speech which we derive from BitPar-parses of the training data: for example *“kaufte<VVFIN>”* (= “bought”, VVFIN = finite verb form of a full verb) or *“haben<VAINF>”* (= “to have”, VAINF = infinite verb form of an auxiliary verb). As described in Chapter 7, particle verbs are split, but their verbal part remains fully inflected.

Prepositions Some German prepositions require the noun to be of a specific *case*, for example *“für”* (= “for”) which occur always with nouns in accusative case. Others can require different cases, depending on the context *“in”* (= “in” in a directive sense with accusative vs. in a local sense with dative). The case feature is left attached to the preposition as we expect the translation model to learn the usage of the grammatical cases from the source language and the context of prepositions in the target language.

gloss	word	feature 1	feature 2	...	feature n	decision
it	Es	<DET>	100,000	...	0	NO
gives	gibt	<VVFIN>	10,000	...	0	NO
children	Kinder	<NN>	10	...	1	YES
punch	Punsch	<NN>	1,000	...	0	NO
and	und	<KON>	100	...	0	NO
gingerbread	Lebkuchen	<NN>	10	...	1	YES
hearts	Herzen	<NN>	10,000	...	0	NO
.	.	<$.>	UNDEF	...	UNDEF	NO

Table 11.2.: Illustration of the CRF training format for the compound merging prediction in the sentence *"Es gibt Kinderpunsch und Lebkuchenherzen."* (= "there is punch for children and gingerbread hearts.").

Note that all German SMT training data is transformed into this underspecified representation prior to training. This will thus be the format into which the English source language is translated.

11.2. Conditional Random Fields

In the past, *Conditional Random Fields* (CRFs) have been used for different kinds of tagging sequential data, e.g. part-of-speech tagging (Lafferty et al., 2001), identification of proteins (McDonald and Pereira, 2005), or compound splitting (Dyer, 2010). In contrast to Hidden-Markov-Models, which are limited to the n-gram feature space, a CRF makes a prediction over the whole sequence. CRFs are thus a suitable model for feature-rich compound merging and inflection prediction.

CRFs for Compound Merging Stymne and Cancedda (2011) train a binary CRF classifier on a split representation of the training data: each word is labeled "YES" if it should be merged with the next word into a compound or "NO" if it should not be merged. A simplified[68] illustration of the CRF training data format is given in Table 11.2. Note that there is one word per line, a number of features assigned to each word and that sentence borders are indicated through empty lines. The rightmost column "decision" indicates whether or not a word should be merged with the next one into a compound.[69]

[68] To enhance readability we show full surface forms of the words instead of the underspecified representation as introduced in the previous Section 11.1.

[69] This also applies for n-ary compounds (with $n > 2$)

We use first order CRFs which can look one word further and one word back for the decision. The features we use are either character strings, frequency-based scores or boolean variables; we will introduce them in detail in Section 11.3 below. Irrespective of the feature type, all values are considered strings in CRFs. This is straightforward and desired for the string features and the boolean variables, but the frequency-based scores require some adaptation: we follow Stymne and Cancedda (2011) and bin the frequency-based scores according to powers of ten. To give an example, a word that has occurred between 1 and 10 times will get a feature value of 10, whereas a word that occurred 11 to 100 times will get a feature value of 100, and so on. This allows us to abstract over the frequency counts of the training data. Otherwise, only words occurring *exactly* as many times as the words in the training data would benefit from frequency-based features.

Training vs. Testing The CRFs are trained on the German section of the parallel training data:[70] whenever a compound is split, its former modifiers are labeled with "YES". These merging decisions will then be learned by the CRF model. However, while the models are trained on clean and fluent German text, they will later be applied to predict merging decisions on disfluent MT output. As we use the surface strings of the words as features, this leads to the following issues: on the one hand, the CRF model mirrors the training data in that it learns all common compounds but its performance on word combinations that have not occured in the training data is less reliable. However, these are exactly the compounds we wish to produce with our approach. On the other hand, as SMT output is less fluent than the training data, the lexicalised sequences will differ substantially from the training sequences, making CRF decisions more difficult. We address this issue in a number of experiments where we use a drastically reduced feature set without surface forms or POS. This allows the model to abstract away from the training data and make better predictions on unseen word combinations, i.e. produce more correct compounds that have not been seen in the parallel training data (cf. Section 13.3).

Stymne and Cancedda (2011) dealt with this by translating all parallel training data with a standard SMT system trained on the same training data set. The compound merging CRFs were then trained on this *noisified* version of the training data, as this was more similar to real SMT output, than in its original version.

[70] We use data of the WMT 2009 shared task on statistical machine translation (∼40 million words).

11.3. Prediction of Merging Points

Compound Merging is challenging because not all two consecutive words that could theoretically be merged into a compound **should** also be merged. This decision does not only depend on the words, but often also on their context.

In the following, we motivate and describe all features we used for CRF training in more detail: target-language features, most of which we adapted from Stymne and Cancedda (2011) are the subject of Section 11.3.1, while the new source-language features we use are presented in Section 11.3.2.

11.3.1. Target-Language Features

This first set of features is derived from the target language context of each word. We marked the features we adapted from Stymne and Cancedda (2011) with a superscript "SC", modified features with "SC+" and new original features with "NEW". All frequency-based features are derived from a large text collection[71] which was transformed to the underspecified representation but without splitting compounds. Each feature description concludes with a concrete example, taken from Table 11.2 above.

Lemma^{SC} The training text itself is used in its underspecified representation (= compounds are split, all words are lemmatised). This allows the CRF model to learn and memorise common German compounds that have occured in the training data. An example value of this feature is ***"Kind<NN><Neut><Sg>"***.

Part-of-speech^{SC} Intuitively, some POS-patterns are more common to occur in German compounds than others. The majority of German compounds are noun+noun compounds, of the kind *"Kind$_{NN}$" + "Punsch$_{NN}$" = "Kinderpunsch$_{NN}$"* (= "punch for children"), but there are also other productive POS-patterns like adjective+noun (e.g. *"Gebraucht|wagen"* = "used car") or noun+adjective (e.g. *"licht|durchflutet"* = "flooded with light"), to name only a few. For this feature, we use POS-tags derived from Bitpar Parses (Schmid, 2004) of the German training data. These belong to the Stuttgart-Tübingen Tagset, STTS.[72] An example value of this feature is ***"<NN>"***.

[71] Here, we use the monolingual training data of the WMT 2009 shared task on statistical machine translation, ~146 million words

[72] `http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html`

Bigram^{SC} This feature indicates how often the current word has occurred in a bigram with the next word in a huge text collection. Assume that the bigram *"es gibt"* (= "there is") occurred 67,943 times in this data set. Recall from above that such frequency-based feature values are packed into bins, in order to allow for more abstraction over the training data. Thus, the example value annotated to the first word *"es"* is then ***"100,000"***.

Merged^{SC} This is the counterpart to the preceding *bigram* feature. It indicates how often the current word occurred in a compound together with the next word in an underspecified text collection in which compounds remain unsplit. A word usually gets a high bigram score **or** a high merged score, with the respective lower score often equaling zero. These two features alone are thus important indicators for the compound merging decision. An example value for *"es"* in Table 11.2 above is ***"0"***, as *"es"* never occurred in the modifier position of a compound with the next word *"gibt"*, whereas for *"Kinder"* (= "children") it is ***"1,000"***, assuming that *"Kinder"* occurred between 101 and 1,000 times in a compound with the following word *"Punsch"* (= "punch").

Modifier^{SC} As Baroni et al. (2002) points out, some words are more likely to occur in modifier or head position of a compound than others. For the *modifier* feature, we counted how often the current word has occured in the modifier position of a compounds in the whole text, i.e. not only with the following word (in contrast to the previous merge feature). This can indirectly serve as an indicator for the productivity of a word: how likely it is that a merge with the next word will result in a sound compound. We modeled this idea of productivity more explicitely in the *productivity* feature as described below. An example feature value for *"Kinder"* (= "children"), assuming that it occured in a total of 35,972 times in modifier position of (different) compounds is thus ***"100,000"***.

Head^{SC} The head feature is the counterpart to the *modifier* feature: it indicates the frequency of occurrence of the present word in the head position of compounds throughout the whole text collection. An example feature value for *"Herzen"* (= "hearts") is ***"100"***.

Modifier vs. Simplex^{SC+} Stymne and Cancedda (2011) used this feature to compare the frequency of the current word in modifier position of any compound with its frequency as a simplex word in the text collection. They used four different groups, namely i) f(modifier) = 0 = f(simplex), ii) f(modifier) < f(simplex), iii) f(modifier) = f(simplex), iv) f(modifier) > f(simplex) in their original implementation. However, we modified this feature to represent the ratio of the respective frequencies instead: an example value of ***"10"*** indicates that the word occurred 10 times more often as a modifier than it did as a simplex word in our text collection.

Head vs. Simplex^{SC} Again, this is the counterpart to the previous *modifier vs. simplex* feature. We modified it to indicate the ratio of a word's frequency in head position of a compounds with its frequency as a simplex word in the same text collection. Assume that *"Kinder"* occurred 100 times in head position of a compound, but with a frequency of 10,000 as a simplex word. The *head vs. simplex* feature value would then be ***"0.01"***.

Additional Features of SC (not used) In addition to these features, Stymne and Cancedda (2011) also used character n-gram features to capture morphological transformations around the merging point. This was helpful in their case as they did not reduce the compound modifiers, but left them as they were (e.g. filler letters remained attached). In contrast, we use lemmatised representations of compound modifiers and take neccessary morphological transformations from the output of the rule-based morphology. Our approach would thus not benefit from such character n-gram features.

Productivity^{NEW} Berton et al. (1996) use the productivity of compound parts as an indicator to split compounds. We use it here in the opposite direction: the more productive a modifier is found to be, the more likely it should be merged with the next word.[73] The *productivity* feature shows how diversly a modifier can be merged with different head types. In contrast to the other frequency-based features, it is based on type counts (instead of token counts). The *productivity* feature is related to the *modifier* feature presented above. First, occurrences of the current word in modifier position of compounds are collected, but then a unique sort according to the different head types is performed and the result is binned into a magnitude of ten (as described for the previous features). An example feature value of ***"10,000"*** for *"Kinder"* indicates a high productivity and is thus a strong indicator to merge the current word with the next word into a compound.

[73] This is in line with observations made by Baroni et al. (2002).

English Pos	**English parse**	**German (split)**
NN+NN	(NP(NN traffic) (NN accident))	Verkehrs Unfall
NN+P+NN	(NP(NN Museum) (PP(IN of) (NN Arts)))	Kunst Museum
ADJ+NN	(NP(ADJ protective) (NN measures))	Schutz Maßnahmen
V-ing+NN	(NP(VBG developing) (NNS nations))	Entwicklungs Länder

Table 11.3.: Common POS-patterns and syntactic constructions of English compounds and their translation into a simplified split German representation.

11.3.2. Source-Language Features

The target-language features presented in the previous section are derived from the often disfluent German SMT output, which might hurt their reliability. In order to decide whether a current German word should be merged with the subsequent word in order to form a compound, we thus additionally use features projected from the English source language input, which is coherent and fluent. To our knowledge, we are the first ones who use such source-language to predict merging decisions for productive compounding. Our source-language features build on the assumption that an English compound should be translated into a compound in German, too. They might thus help merging decisions in unclear cases where the target language frequencies alone do not give a clear indication. For example, because they are very low (only a few occurrences of a word as a modifier, not clearly indicating that the word represents a modifier in the present case), or they have occurred with roughly the same frequency in any position of the word (compound modifier vs. head or simplex word). We use three different types of source language features: i) Pos-tag of the English word, ii) syntactic features, and iii) alignment features, all of which we motivate and describe in the remainder of this section. The superscript "EN" indicates that the features are derived from the English source language. Note that all source-language features (except the POS-feature) are realised as boolean variables: they are either applicable or not.

Part-of-speech Feature^{EN} Recall from the previous section on target-language features that the POS of a word is a helpful indicator for the merging decision as some POS-patterns are more likely to occur in a compound than others. Not only the POS of the current German word but also the POS from the English input word to which it has been aligned to should be taken into consideration: as Corston-Oliver and Gamon (2004)

should <u>not</u> be merged:	**should be merged:**
darf ein **kind punsch** trinken? “may a **child** have a **punch**?”	jeder darf **kind punsch** haben! “everyone may have punch for children!”
(TOP(SQ(MD May) (**NP**(DT a)(NN **child**)) (VP (V have) (**NP** (DT a)(NN **punch**)))(. ?)))	(TOP(S(NP(NN Everyone))(VP(MD may)(VP(V have) (**NP**(NP(NN **punch**))(PP(IN for)(NP(NN **children**))))))(. !)))

Figure 11.3.: Example of how the English sentence structure helps the merging decision of *“Kind”* and *“Punsch”*.

report, English noun groups are generally translated into German noun compounds and Rackow et al. (1992) describe that this also holds for the opposite direction. We use the POS of the English words to learn (un)usual POS-combinations. In Table 11.3, we give some examples of typical POS-patterns of English compounds (leftmost column) together with their translation into German[74] (rightmost column). The POS-feature is independent of the syntactic structure of the source sentence (as given in the middle column of Table 11.3), but we will use it to derive the syntactic source-language features.

Syntactic Features^{EN} For the syntactic features, not only the current German word, but also the next German word is taken into account: knowing that both are aligned to two English words that fit into a typical English compound pattern is a strong indicator that these German words should be merged into a compound. We parsed the English training data with the constituent parser Charniak and Johnson (2005), using a model trained on the standard Penn Treebank sections. In the parsed data, we check four different English compound patterns: *noun+noun*, *noun+preposition+noun*, *adjective+noun* and *gerund-verb+noun* (see Table 11.3 for examples).

A contrastive example for the English *noun-preposition-noun* pattern is given in Figure 11.3, again in a simplified format to enhance readability: in the left example, *“Kind”* and *“Punsch”* are the subject and the object of the sentence, respectively, and thus should not be merged. The syntactic structure of the source sentence reveals that “child” and “punch” do not occur within one common noun phrase, which is a strong indicator **not** to merge them. In contrast, “child” and “punch” fit into the “noun-preposition-noun” compound pattern in the right example. It can be seen from the syntactic structure that they are both covered by one common noun phrase. Here, the two German words *“Kind”* and *“Punsch”* should definitely be merged into *“Kinderpunsch”*.

[74] Note that the German examples are given in a simplified format to enhance readability.

Alignment Features^{EN} Finally, we use alignment features to promote the merging of compounds which should not have been split in the first place. This concerns compounds that are translated into one single word in English, e.g. because they are lexicalised in English as in the case of *"Blumen|topf"* (= "flowerpot") or they have a non-compositional semantics, e.g. *"Hecken|Schütze"* (= "sniper", lit.: "hedge|shooter").[75] This 1:n asymmetry in the word alignment from English to German is used to identify them.[76] We check whether the current German word and the next word are aligned to the same word in the English source sentence:

i) neither the German words nor the English word is aligned to any other word
ii) same as i) but the English word contains a dash
iii) the German words and the English word may also be aligned to other words
iv) based on iii) but the English word contains a dash.

11.4. Inflection Prediction

Besides the CRF models we use for compound merging, we also use already existing CRF models for inflection. This section gives a brief summary of relevant features for the inflection prediction models, trained by Marion Weller, Aoife Cahill, and Alexander Fraser. These models have been published previously in Weller (2009) and Fraser et al. (2012), and the authors kindly made them available for this thesis.

An example is given in Table 11.4: the underspecified representation of the two nouns *"Socken"* (= "socks") and *"Teddy"* (= "teddy") shows that *gender* and *number* are word-inherent morphological properties.[77] These will thus not be predicted for the nouns but they are still part of the feature vector for training purposes. Their values are propagated to related constituents in order to model the agreement within phrases. This happens prior to training using parse trees obtained from BitPar (Schmid, 2004). Moreover, the parse trees are also used for the context-sensitive disambiguation of the other properties *case* and *definiteness*, which are not word-inherent to nouns and have to be determined for the training of CRF models. A testing time, such context-dependent properties will be predicted by the CRF model based on the context. All properties that will be predicted

[75] See also Section 5.1.1 above.

[76] Popović et al. (2006) used n:1 alignments for source language word packing of English words in the opposite translation direction.

[77] We usually call them morphological features, but in order to not confuse them with CRF features in this section, we renamed them to morphological properties.

gloss	**lemma**	**pos**	**gender**	**number**	**case**	**definite**
the	die<ART><Def>	ART	masc	pl	nom	weak
little	klein<ADJ><Pos>	ADJ	masc	pl	nom	weak
socks	Socke<+NN><Masc><Pl>	NN	masc	pl	nom	weak
are	sind<VVFIN>	VVFIN	X	X	X	X
for	für<APPR><Acc>	APPR	X	X	acc	X
a	ein<ART><Indef>	ART	fem	sg	acc	strong
little	klein<ADJ><Pos>	ADJ	fem	sg	acc	strong
teddy	Teddy<+NN><Fem><Sg>	NN	fem	sg	acc	strong
.	.<$.>	<$.>	X	X	X	X

Table 11.4.: Illustration of the training data format for the 4 inflection prediction CRFs for the sentence *"Die kleinen Socken sind für einen kleinen Teddy"*. Features that will be predicted by the respective models are highlighted .

are highlighted, while all word-inherent properties remain with white background in Table 11.4. As can be seen, the *case* feature remains attached to prepositions (here: *"für"* = "for") and *definiteness* is a word-inherent part of the determiners (here: *"die"* = "the", *"ein"* = "a"). Their values are thus important indicators for the CRF to determine the correct *case* and *definiteness* properties for the whole noun phrase at testing time.

Training of one single CRF for all properties to be predicted was not tractable, we thus trained four separate CRFs instead: one for each context-sensitive feature that is to be predicted from the underspecified SMT output: *gender*, *number*, *case* and *definiteness*. Note that not all of these properties are predicted for all word classes. As shown in the previous example, only *case* and *definiteness* are to be predicted for nouns, as *gender* and *number* are word-inherent properties. In contrast, all properties need to be predicted to properly re-inflect adjectives and nothing at all is predicted for prepositions (here: *"für"* = "for"), as they are not inflected in German and keep their *case* feature in the underspecified representation. The features *lemma* and *pos* are used in all models, but the other features (given in Table 11.4) are only used to train their respective CRF. For each of these features,[78] a ±5-gram sequence window of the current position is used in order to allow the CRF model to learn agreement sequences. In contrast to the compound merging CRF, where the model predicts a binary decision (whether or not to merge a current word with the next word cf. Section 11.3), the re-inflection CRFs predict multiple values for each of the features: *gender* = masculine, feminine, neuter; *number* = singular, plural; *case* = nominative, genitive, dative, accusative; *definiteness* = strong, weak.

[78] Except for *pos*, for which a ±7-gram window in both directions is used.

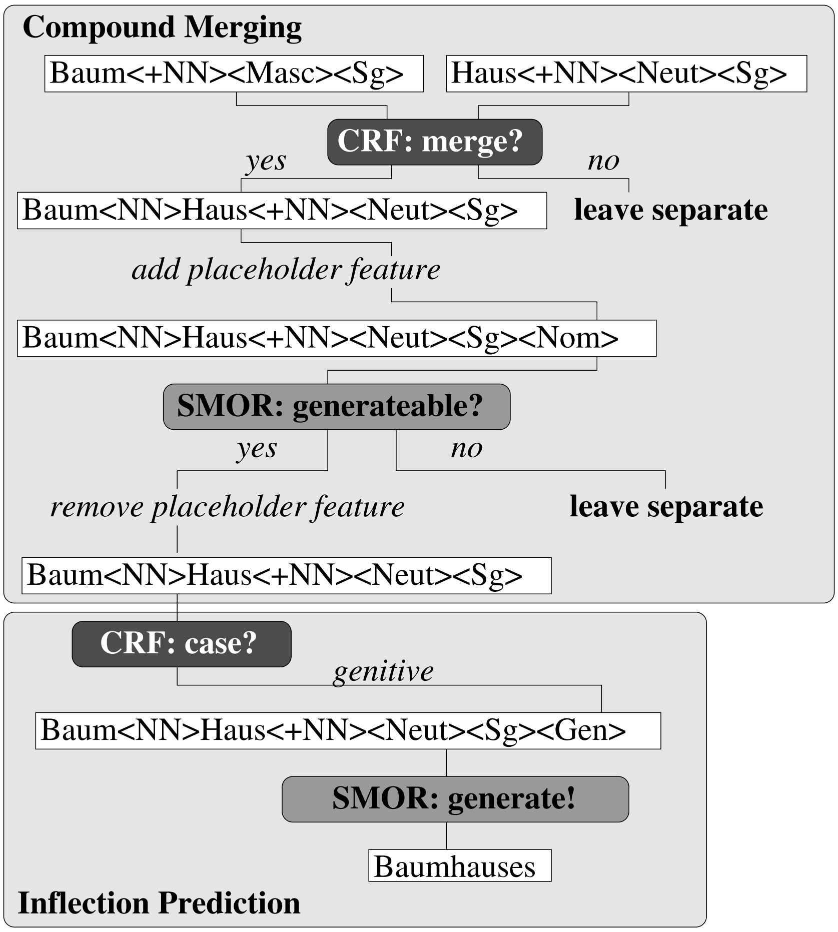


Figure 11.4.: Illustration of the morphological generation process for *"Baumhaus"* (= "tree house"): compounds are first merged and then re-inflected.

11.5. Morphological Generation

In the previous sections, we presented the underspecified representation and gave details on how Crf models are trained for the prediction compound merging and inflection. This section deals with the re-transformation of the underspecified representation into fluent, fully inflected German by taking into account the predictions of the respective Crf models. The process of compound merging is described in Section 11.5.1 and re-inflection in Section 11.5.2. Finally, after re-inflection, portemanteaus remain to be merged in Section 11.5.3.

11.5.1. Compounds

The compound merging procedure is illustrated in Figure 11.4. The CRF model indicates which words should be merged into a compound, but before two (or more) words are finally merged into a compound, the candidate must undergo a sanity check through SMOR. Words that are not generatable by SMOR are either not covered by its lexicon or they violate German word formation rules (e.g. certain POS classes cannot occur within compounds). We thus restrict compound merging to words that can be generated by SMOR. For this *sanity-check generation*, all features except the POS must be removed from the modifier(s) (here: *"Baum<NN>"*) and the head must be enriched with a placeholder feature (here: *case* = <Nom>), as SMOR requires fully decorated feature representations as input for generation. If the test is successful, the word will be re-transformed into the underspecified representation by removing the placeholder feature. The compound is now merged and ready for re-inflection, which will be described in the next section. If not, the word is left in its split representation.

Particle Verbs Verbs remain fully inflected throughout the whole translation pipeline. It thus merely makes a difference where to restore them. However, as the original re-inflection CRFs by Weller, Cahill, and Fraser have been trained on unsplit text, we merge particles and verbs prior to nominal re-inflection, in order to prevent undesired side-effects. German particle verbs are a closed (only sparsely productive) word class, but there is a considerable number of different combination possibilities between particles and fully inflected verb forms. Instead of using a list-based approach to restore them (which we will use for portemanteau merging, see Section 11.5.3), we rely only on the part-of-speeches of the words to decide whether or not a verb particle and a verb should be merged into a particle verb.

11.5.2. Re-Inflection

After having predicted all relevant features with the four separate re-inflection CRFs, the underspecified representation is enriched and SMOR is used to generate full forms (see Figure 11.4). As previously mentioned, we use an inflection prediction pipeline implemented by Marion Weller, Aoife Cahill and Alexander Fraser, but with a slightly modified generation process for compounds. Recall from Section 2.1 that German compound formation sometimes requires a morphological transformation of the modifier(s): e.g. a filler

no portemanteau merging:							
Das	Haus	,	in	dem	ich	wohne	.
ART	NN	$,	PREP	PREL	PPOS	VVFIN	$.
the	*house*	*,*	*in*	*the/which*	*I*	*live*	*.*

portemanteau merging:						
Ich	wohne	alleine	in	dem	Haus	.
PPOS	VVFIN	ADV	PREP	ART	NN	$.
I	*live*	*alone*	*in*	*the*	*house*	*.*
Ich	wohne	alleine	**im**		Haus	.

Table 11.5.: The POS of the words is taken into account for portemanteau merging: only prepositions + articles are merged.

letter must be introduced as in *"Ort"*+*"Zeit"* = *"Ortszeit"* ("local time"), or deleted as in *"Erde"*+*"Oberfläche"* = *"Erdoberfläche"* ("earth's surface"). We use Smor to generate compounds from a combination of simple words. This allows us to create compounds with modifiers that never occurred as such in the parallel training data. Imagine that *"Ort"* occurred only as compound head or as a single word in the training data. Using Smor for compound generation, we are still able to create the correct form of the modifier, including the required filler letter: *"Orts"*. However, this generation is not always unique: for example, *"Kind"* (= "child") + *"Punsch"* (= "punch") → *"Kinderpunsch"* but *"Kind"* (= "child") + *"Vater"* (= "father") → *"Kindesvater"*. Such over-generations are typical for FST-based morphological analysers like Smor, which have mainly been conceived for analysis and robustness. We gave details on how over-generations can emerge from the way compounding is implemented in Smor in Section 3.2.2. Whenever Smor generates more than one possibility, the original re-inflection pipeline of Weller, Cahill, Fraser chooses the first possibility by default. In contrast, we disambiguate the output using corpus frequencies. We check whether the compound occurred in a large corpus, and if not, we chose the most frequent modifier form, independent of the head type.

11.5.3. Portemanteau Merging

After the whole text has been re-inflected, the portemanteaus, which were split prior to training, have to be restored. There is only a limited number of them, and we thus use a list of 13 portemanteaus[79] and their split representations to produce them. In addition,

[79] Namely: *am, ans, aufs, beim, im, ins, übers, überm, vom, vor, zum, zur.*

the POS tag of the words is taken into account in order to prevent erroneous mergings. The example in Table 11.5 shows that merging *"in"*+*"dem"* into the portemanteau *"im"* should only be performed when *"dem"* is used as a determiner (POS = "ART"). Otherwise, for example when used as a relative pronoun (POS = "PRELS") the two words should be left separated.

Sometimes, the merging of a preposition and a determiner into a portemanteau of our list does not adequately capture the meaning of a sentence, even though the sentence is grammatically correct. However, such exceptions are very rare and it would require a semantic interpretation to capture them.

11.6. Chapter Summary

In this chapter, we presented our compound merging procedure in detail. In order to account for maximal generalisation, our preprocessing of the German data covers not only compound splitting but also lemmatisation. We introduced an underspecified representation of German serving as the training and translation format. After translation, post-processing is required to generate fully specified German from this representation. We make use of conditional random fields to predict compound merging points and nominal inflection. Finally, we use the rule-based morphological analyser SMOR to generate inflected full forms. In the next chapter, we will investigate the accuracy of the compound prediction CRF on held-out data.

12. Compound Prediction Accuracy

In the previous chapter, we presented our compound merging procedure in detail. It makes use of machine learning (conditional random fields, CRFs) to predict whether simple words should be merged into compounds. In the present chapter, we evalute the accuracy of the CRF compound prediction models on clean data, before we include the whole compound processing pipeline into end-to-end SMT experiments in Chapter 13. This allows us to get a first impression on the reliability of the different feature combinations we used to train the models. Accuracies are measured with respect to automatically obtained gold annotations using the *precision*, *recall* and *F-score* metrics. Note that we already published parts of these results in Cap et al. (2014a).

Inflection Accuracy In our compound processing approach, we use CRFs not only to predict suitable merging points of simple words, but also to predict grammatical features like e.g. *case* or *number*. As mentioned earlier, we reuse an inflection handling component developed by Marion Weller, Alexande Fraser and Aoife Cahill. In Weller (2009) and Fraser et al. (2012), the accuracies of different inflection prediction models have already been examined on clean data. The four CRF model cascade we use was found to score highest, namely 94.29% accuracy without compound processing. We adapted their models without any major modifications and thus do no re-calculate clean data accuracies. The only modfication we performed was an improved compound selection in the final morhpological generation process, which is independent of the CRF models themselves. Details are given in Section 11.5.2.

Structure The remainder of this chapter consists of two parts: In Section 12.1, we give details on the experimental settings we used. In Section 12.2 we report on compound prediction accuracies. Section 12.3 briefly summarises this chapter.

12.1. Setup

In this section, we describe the experimental settings we used to evaulate compound prediction accuracies, including the data, the creation of gold annotations, different experiments (in terms of different feature combinations) and how we evaluated the outcome of the predictions against the gold annotations.

Data In order to be able to use source language features for the Crfs, it is necessary to use a parallel text. We use data from the EACL 2009 workshop on statistical machine translation.[80] The compound prediction Crfs are trained on the parallel training data (∼40 million words), but the frequencies of the target language features are derived from the monolingual training data, consisting of roughly 227 million words.

Gold Annotations Starting from the parallel training data, compounds are split using the compound splitting approach described in Chapter 7 and the whole text is transformed into the underspecified representation introduced in Section 11.1. The task of compound merging can be defined as a reversion of compound splitting. The compound splitting decisions are thus stored in the course of splitting and will be learned by the compound merging Crf as merging decisions.

Experiments An overview of the different feature combinations we used is given in Table 12.1. More detailed feature descriptions can be found in Section 11.3 above. In order to motivate our new feature combinations and to make them comparable to previous work, we trained one model (Sc) using only the target-language features described in Stymne and Cancedda (2011). Note, however, that this comparison only concerns the accuracy of the prediction model. We use the same morphology-aware compound splitting approach for all merging experiments, including (Sc). Due to the underspecified representation we use, we excluded the n-gram character features described in Stymne and Cancedda (2011) in our Sc experiment[81]. Besides Sc, we trained four more models based on two different target-language feature sets: one full feature set, with (St) and without (T) source language features and one reduced target language feature set with (Str) and without (Tr) source language features.

[80] `http://www.statmt.org/wmt09`

[81] A closer re-implementation of Stymne and Cancedda (2011)'s work is beyond the scope of this work, as this would include using a different splitting approach, factored SMT, no modifier normalisation, no inflection prediction, and a noisification of the CRF training data.

Feature			**Experiment**				
No	**Short Description**	**Type**	**Sc**	**T**	**Tr**	**St**	**Str**
1SC	underspecified representation of the word	string	X	X		X	
2SC	main Pos of the word	string	X	X		X	
3SC	word occurs in a bigram with the next word	freq.	X	X		X	
4SC	word combined to a compound with the next word	freq.	X	X	X	X	X
5SC	word occurs in modifier position of a compound	freq.	X	X		X	
6SC	word occurs in a head position of a compound	freq.	X	X		X	
7SC	word occurs in modifier position vs. simplex	string	X				
8SC	word occurs in head position vs. simplex	string	X				
7SC+	word occurs in modifier position vs. simplex	ratio		X	X	X	X
8SC+	word occurs in head position vs. simplex	ratio		X	X	X	X
9N	different head types the word can combine with	freq.		X	X	X	X
10E	Pos of the corresponding English word	string				X	X
11E	English noun phrase	bool.				X	X
12E	English gerund construction	bool.				X	X
13E	English genitive construction	bool.				X	X
14E	English adjective noun construction	bool.				X	X
15E	aligned uniquely from the same English word	bool.				X	X
16E	like 15E, but the English word contains a dash	bool.				X	X
17E	like 15E, but not only unique links	bool.				X	X
18E	like 16E, but not only unique links	bool.				X	X

Table 12.1.: Overview of Compound Merging experiments.
Crf features: **SC** = features taken from Stymne and Cancedda (2011), **SC+** = improved versions, **N** = new feature, **E** = features projected from the English input.
Experiments:**Sc** = re-implementation of Stymne and Cancedda (2011), **T**= use full **T**arget feature set, **Tr** = use **T**arget features, but only a **R**educed set, **St** = use **S**ource language features plus **T**, **Str** = use **S**ource language features plus **Tr**

Evaluation We use the tuning set from the 2009 WMT shared task to evaluate the Crf models we trained on the respective training set of WMT 2009. It consists of 1,025 sentences. The gold annotations were obtained in the same way as the training data by remembering split points. The evaluation procedure consists of the following steps:

1. split compounds of the German wmt2009 tuning data set (= 1,025 sentences)
2. remember compound split points and store them as gold annotations
3. predict merging points with Crf models
   → calculate F-scores to indicate Crf prediction accuracies
4. merge predicted words into compounds using SMOR
   → calculate F-scores on how properly the compounds were merged

F-scores are calculated using the following formula: $F = \frac{2*(precision*recall)}{precision+recall}$

	all	compounds				particle verbs		
		all	2 parts	3 parts	4 parts	all	2 parts	3 parts
labels	1,427	1,151	967	172	12	276	154	122
words	1,272	1,057	967	86	4	215	154	61

Table 12.2.: Distribution of merging labels, and words to be merged in the German wmt2009 tuning set.

12.2. Results

In this section, we will report on two different kinds of accuracies of the CRF compound merging models: i) the labeling accuracy which indicates how well the CRF model predicts merging decisions, and ii) the merging accuracy which indicates how many of the compounds in the dataset have been restored correctly. The latter is particularly important for n-ary compounds (with n>2), as more than one correct labeling decision is necessary to generate the compound. In the following, we first give an impression of the distribution of labels and compounds in our dataset. Then, we give some calculation details of the accuracies of our different CRF models and finally discuss these accuracies.

Distribution of Labels Table 12.2 illustrates the distribution of merging labels over the numbers of compounds and particle verbs. This table shows that, for example, 86 compounds consisting of three parts require 172 labels to be predicted correctly in order to be generated accordingly. Moreover, it can be seen that 91% of all compounds in our dataset consist of two parts, 8% consist of three parts and only 1% of four parts.

Details of Calculation We use precision, recall and F-score to report on the accuracies of the CRF models.[82] In Table 12.3 we give results for the five different CRF models we trained, for both the labeling and the merging accuracy respectively. Note that the merging accuracy is based on compounds only, as particle verbs are merged with a rule-based approach and can thus be 100% restored from a clean dataset. For compounds, we first performed an oracle merging with correct labels to measure the accuracy of restoring compounds with SMOR that have previously been split with SMOR, independently of the labeling accuracy. We found that 1,047 of the 1,057 compounds (= 99.05%) could

[82] The formula for the calculation of F-scores is given on the bottom of the previous page. See Section 8.1.2 (page 94) for details on how precision and recall are calculated.

(a) Labeling Accuracy: 1,427 merge points to be found.

exp	**to be**	**all**	**correct**	**wrong**		**precision**	**recall**	**f-score**
	labeled	**labeled**	**labeled**	**labeled**	**not labeled**			
SC	1,427	1,360	1,263	97	164	92.87%	88.51%	90.64%
T	1,427	1,325	1,254	71	173	94.64%	87.88%	91.13%
TR	1,427	1,244	1,159	85	268	93.17%	81.22%	86.78%
ST	1,427	1,329	1,259	70	168	94.73%	88.23%	91.36%
STR	1,427	1,301	1,200	101	227	92.24%	84.09%	87.98%

(b) Merging Accuracy: 1,047 compounds to be merged.

exp	**to be**	**all**	**correct**	**wrong**			**precision**	**recall**	**f-score**
	merged	**merged**	**merged**	**merged**	**not merged**	**faulty**			
oracle	1,057	1,047	1,047	0	10	0	100%	99.05%	99.52%
SC	1,047	997	921	73	121	3	92.38%	88.13%	90.21%
T	1,047	979	916	59	128	4	93.56%	87.40%	90.38%
TR	1,047	893	836	52	204	5	93.62%	80.00%	86.27%
ST	1,047	976	917	55	126	4	93.95%	87.58%	90.66%
STR	1,047	930	866	58	172	6	93.12%	82.95%	87.74%

Table 12.3.: Overview on compound prediction accuracies of the different models. **Sc** = re-implementation of Stymne and Cancedda (2011), **T**= use full **T**arget feature set, **Tr** = use **T**arget features, but only a **R**educed set, **St** = use **S**ource language features plus **T**, **Str** = use **S**ource language features plus **Tr**.

be restored.[83] The merging accuracies given in Table 12.3 are calculated with respect to this upper bound.

Discussion While the re-implementation of Stymne and Cancedda (2011) scores highest in recall, our models (even the ones with the reduced feature sets, **Tr** and STR) tend to outperform it in terms of precision. Overall, it can be seen from Table 12.3 that using more features (**Sc**→**T**→**St**) is favourable in terms of precision and overall accuracy and the positive impact of using source language features is clearer for the reduced feature sets (**Tr** vs. **Str**). These observations are similar for both kinds of accuracies. For example, the use of source language features in the reduced setting (**Str**) leads to 30 more correctly merged compounds than in the contrastive experiment without source language features (**Tr**), namely 866 vs. 836.

However, these accuracies only somewhat correlate with SMT performance: here, the models are trained and tested on clean, fluent German language, but later they will be applied to disfluent SMT output and might thus lead to different results there.

[83] The ten missing compounds cannot be generated due to minor formatting issues, e.g. mismatching POS tags between the parser and SMOR, numeric compounds, or casing mismatches.

exp	**WER**	**SER**
Sc	0.10%	2.64%
T	0.15%	3.99%
Tr	0.52%	12.74%
St	0.12%	3.09%
Str	0.40%	9.90%

Table 12.4.: Crf model training error rates. WER = word error rate, SER = sentence error rate; **Sc** = re-implementation of Stymne and Cancedda (2011), **T**= use full **T**arget feature set, **Tr** = use **T**arget features, but only a **R**educed set, **St** = use **S**ource language features plus **T**, **Str** = use **S**ource language features plus **Tr**.

Nevertheless, the accuracies of the Crf models on clean data can indicate whether low training error rates of Crf model training are a sign of high performance of the model or whether these training error rates are a consequence of model over-fitting to the clean training data, which is not desirable for our application. The word error rates (WER) and sentence error rates (SER) of the Crf training logs are given in Table 12.4. Comparing these error rates to the accuracies in Table 12.3, we find that the model with the lowest training error rates, **Sc** is **not** the most accurate one. This is an indicator that the CRF model may have overfit to the training data.

12.3. Chapter Summary

In this chapter, we investigated the accuracies of the Crf-based compound merging models. We calculated f-scores based on a held-out dataset with automatically obtained gold annotations and found that our feature combinations mostly outperform previous work in terms of precision. Moreover, the results showed a small improvement whenever source language features were used in addition to target language features. In the next chapter, we will integrate all of the Crf-models for compound prediction examined here into end-to-end SMT.

13. SMT Evaluation: English to German

In previous chapters, we introduced our compound merging procedure, which includes inflection handling of the target language output (Chapter 11) and measured the accurracies of different compound prediction models (Chapter 12). In this chapter, we integrate our compound merging procedure, combined with inflection handling into English to German SMT. We compare it to a system handling inflection but not compounds and to a raw baseline with neither compound nor inflection handling. Moreover, we compare the impact of different feature combinations of the compound merging models. Both the results and addtional manual evaluations show that our compound processing has a positive effect on translation quality. Our systems produce more German compounds that are correct translations of the English input than the baselines systems, and a considerable number thereof has not occured in the parallel training data.

Motivation One of the motivations behind compound processing in SMT is to enable the translation of words that have not occured in the training training data by decomposing them into parts that have occurred in the training data. We have already shown in Chapter 9 above, that compound splitting improves the overall translation quality of a German to English SMT system, but also that the number of unknown words decreased by 30%.[84] In the opposite direction however, translating from English to German, compound processing is much more challenging: not only does translation quality depend on adequate compound splitting, but also on the correct merging of simple words (back) into compounds after translation, thereby potentially creating new compounds that have not occured in the parallel training data. Moreover, the inflectional ending of the merged compound must agree within its target language sentence context.

[84] See Table 9.3 in Section 9.2.2 for details.

Previous Work For compound merging in Smt, Stymne and Cancedda (2011) present an approach based on POS-based splitting and CRF-based merging. However, compound merging is only applied on held out data. It is not integrated into end-to-end Smt. The inflection prediction models we use are inspired by Toutanova et al. (2008), who worked on Russian and Arabic inflection.

To our knowledge, we were so far the only ones to compound merging and inflection prediction in Fraser et al. (2012). However, compound merging was at that time restricted: compound modifiers were not reduced to their base forms (= blocks full generalisation over all former compound parts) and the morphological generation of merged compounds was performed with a list-based approach (= dependency on the list coverage). The compound merging procedure we present in this thesis, which allows for full generalisation and generates all compounds directly from the rule-based morphology has been published in Cap et al. (2014a), including some of the results given in this chapter. More previous work is to be found in Chapter 14.

Structure The remainder of this chapter is structured as follows: we give details of our Smt system in Section 13.1 and Smt results in terms of Bleu scores in Section 13.2. In Section 13.3, we report on a detailed manual evaluation of the compounds that have been produced by our best performing system in great detail. We investigate the human perception of translation quality with and without compound processing in Section 13.4. In Section 13.5, we compare our results to translations obtained from an external state-of-the-art system (Google Translate). We summarise the chapter in Section 13.6.

13.1. Experimental Setting

In the following, we give details on the experimental setup for our English to German Smt experiments. Technical details of the system are given in Section 13.1.1 and an overview of the different compound merging strategies is given in Section 13.1.2.

13.1.1. Translation System

In order to examine the effects of compound merging and inflection handling in Smt, we train Smt systems on three different data representations: i) a Raw baseline without compound or inflection processing, ii) a baseline without compound processing but with inflection handling (UNSPLIT) and iii) several systems with compound processing and inflection handling (SPLIT*). The SPLIT* systems are all trained identically, with identical compound splitting, differing only in the way compounds are merged after translation. In general, we use the same dataset and tools as we did for the opposite translation direction in Chapter 9.

Data We use data from the shared task of the EACL 2009 workshop on statistical machine translation.[85] The training data consists of ∼1.5 million parallel sentences (∼40 million words). It is composed of the proceedings of the European parliament debates (cf. EUROPARL corpus, version 4, (Koehn, 2005)) and also some news texts. Sentences longer than 40 words were removed from the training corpus. We use 1,025 sentences for tuning and 1,026 sentences for testing. All data was lower-cased and tokenised, using the shared task tokeniser. For the compound processing systems (SPLIT*), the German sections of the bilingual training data was split using SMOR (see Chapter 7 for details). For the inflection handling system (UNSPLIT) and for the compound processing systems (SPLIT*), respectively, the data is then transformed into a lemma-like underspecified representation (see Section 11.1 for details).

Language Model Based on the German monolingual training data of the shared task (containing roughly 227 million words), we trained different language models for the three different representations (RAW, UNSPLIT, SPLIT*). All language models are based on 5-grams and trained using the SRILM toolkit (Stolcke, 2002) with Kneeser-Ney smoothing. We then use KenLM (Heafield, 2011) for faster processing.

Translation Model We used the multithreaded GIZA++ toolkit for word alignment (Och and Ney, 2003; Gao and Vogel, 2008). For translation model training and decoding, we use the Moses toolkit (Koehn et al., 2007) to build phrase-based statistical machine translation systems.[86] For the RAW baseline, the UNSPLIT baseline and all SPLIT* systems we built a separate system. We did so by closely following the instructions of the shared task[85], using only default parameters.

[85] `http://www.statmt.org/wmt09/translation-task.html`

[86] The Moses toolkit can be obtained from `http://www.statmt.org/moses`; we used version 1.0.

Tuning For tuning of feature weights we ran Minimum Error Rate Training (Och, 2003) with batch-mira (Cherry and Foster, 2012) and *'–safe-hope'* until convergence (or maximal 25 runs), optimising BLEU scores (Papineni et al., 2002). For the SPLIT* systems, the CRF-based merging of compounds was integrated into each iteration of tuning and scored against a lemmatised but not split version of the tuning reference.[87] The same reference was used for the UNSPLIT system. In contrast, the output of the RAW system was scored against the original (tokenized, lowercased) tuning reference. Due to the underspecification format (leading to more reference matches), the tuning scores of the UNSPLIT and SPLIT* systems are generally higher than for RAW. However, this effect gets lost at testing time, where BLEU is scored against fully inflected text.

Testing The outputs of the UNSPLIT and SPLIT* systems require post-processing before they are evaluated using BLEU. For the compound processing systems (SPLIT*), testing consists of:

1. translation into the split, underspecified German representation
2. compound merging using CRF models to predict recombination points
3. re-inflection using the CRF models of Weller (2009) and Fraser et al. (2012)

The same applies to the UNSPLIT system, but without the compound merging step (step 2). After decoding and post-processing, the output texts of all systems were automatically recapitalised and detokenised, using the tools provided by the shared task. For translation quality estimation, we calculated Bleu (Papineni et al., 2002) scores using version 11b, see Section 4.2.3 for more details on how exactly BLEU scores approximate translation quality. We did significance testing using pair-wise bootstrap resampling with sample size 1,000 and a p-value of 0.05.[88]

13.1.2. Experiments

Raw Baseline RAW denotes a very simple contrastive system which we built closely following the instructions of the shared task for the construction of a baseline system.[89] For each step, we used only default parameters and we did not perform any kind of pre- or post-processing on any of the underlying datasets.

[87] In contrast, El Kholy and Habash (2010) tuned against a split and lemmatised Arabic reference set.

[88] The code can be obtained from `http://www.ark.cs.cmu.edu/MT`

[89] These can be obtained from `http://www.statmt.org/wmt09/baseline.html`

Feature			Experiment				
No	**Short Description**	**Type**	**Sc**	**T**	**Tr**	**St**	**Str**
1SC	underspecified representation of the word	string	X	X		X	
2SC	main Pos of the word	string	X	X		X	
3SC	word occurs in a bigram with the next word	freq.	X	X		X	
4SC	word combined to a compound with the next word	freq.	X	X	X	X	X
5SC	word occurs in modifier position of a compound	freq.	X	X		X	
6SC	word occurs in a head position of a compound	freq.	X	X		X	
7SC	word occurs in modifier position vs. simplex	string	X				
8SC	word occurs in head position vs. simplex	string	X				
7SC+	word occurs in modifier position vs. simplex	ratio		X	X	X	X
8SC+	word occurs in head position vs. simplex	ratio		X	X	X	X
9N	different head types the word can combine with	freq.		X	X	X	X
10E	Pos of the corresponding English word	string				X	X
11E	English noun phrase	bool.				X	X
12E	English gerund construction	bool.				X	X
13E	English genitive construction	bool.				X	X
14E	English adjective noun construction	bool.				X	X
15E	aligned uniquely from the same English word	bool.				X	X
16E	like 15E, but the English word contains a dash	bool.				X	X
17E	like 15E, but not only unique links	bool.				X	X
18E	like 16E, but not only unique links	bool.				X	X

Table 13.1.: Overview of Compound Merging experiments.
Crf features: **SC** = features taken from Stymne and Cancedda (2011), **SC+** = improved versions, **N** = new feature, **E** = features projected from the English input.
Experiments:**Sc** = re-implementation of Stymne and Cancedda (2011), **T**= use full **T**arget feature set, **Tr** = use **T**arget features, but only a **R**educed set, **St** = use **S**ource language features plus **T**, **Str** = use **S**ource language features plus **Tr**

Unsplit Experiment This system is a re-implementation of the inflection handling system without compound processing as described in Fraser et al. (2012). As inflection handling is an integral part of the compound processing systems, the Unsplit system allows us to to measure the effect of compound processing in isolation.

Compound Merging Experiments (Split*) We compare the influence of different compound merging strategies in end-to-end Smt. An overview of the different feature combinations we use for the prediction of merging points, together with the names of the respective experiments is given in Table 13.1.[90] More detailed feature descriptions can be found in Section 11.3.

[90] This table is a reproduction of Table 12.1.

experiment	**Bleu scores**		
	mert.log	**Bleu**	**RTS**
Raw	14.88	14.25	1.0054
Unsplit	15.86	**14.74**	0.9964
Split-Sc	15.44	14.45	0.9870
Split-T	15.56	14.32	0.9634
Split-St	15.33	14.51	0.9760
Split-Tr	15.24	14.26	0.9710
Split-Str	15.37	**14.61**	0.9884

Table 13.2.: English to German Smt results on data from the WMT 2009 shared task. Tuning scores (mert.log) are on merged but uninflected data (except RAW). With **Sc** = re-implementation of Stymne and Cancedda (2011), **T**= use full **T**arget feature set, **Tr** = use **T**arget features, but only a **R**educed set, **St** = use **S**ource language features plus **T**, **Str** = use **S**ource language features plus **Tr**; **RTS**: length ratio; **bold** face indicates a statistically significant improvement over the Raw baseline, Sc, T and Tr.

13.2. Results

The results are given in Table 13.2. It can be seen that both, the Unsplit and all Split systems outperform the Raw baseline. For Unsplit and Split-Str (source language and a reduced set of target language features) the improvements over Raw are statistically significant. Moreover, they also significantly outperform all other systems, except Split-St (full source and target language feature set). The difference between Split-Str (14.61) and the Unsplit baseline (14.74) is **not** statistically significant.[91]

These observations deviate from the merging accuracies on clean data, which we investigated in Section 12.2 (p. 169) above. There, we concluded that *"using more features (***Sc**→**T**→**St***) is favourable in terms of precision and overall accuracy and the positive impact of using source language features is clearer for the reduced feature sets (* **Tr** *vs.* **Str***)"*. In terms of translation quality, we can see from the Bleu scores in Table 13.2 that the merging accuracies are only partly correlated with Smt performance: while more features lead to higher Bleu scores for Split-St vs. Split-Sc (14.51 vs. 14.45), the use of our target language features alone (Split-T) results in a lower score (namely 14.32), despite the use of the additional productivity feature. On the other hand, the second conclusion drawn from the clean data accuracies still holds: the effect of using

[91] Indirectly, the results from the 2014 shared task (Cap et al., 2014b) confirm this difference not being significant. There, the compound processing system scores higher: CimS-RI (= Unsplit) yields 17.75 and CimS-CoRI (= Split-Str) yields 17.87, but again, the difference is not significant.

experiment	**#compounds found**			
	all	**ref**	**new**	**new***
RAW	646	175	n.a.	n.a.
UNSPLIT	661	185	n.a.	n.a.
SPLIT-SC	882	241	47	8
SPLIT-T	845	251	47	8
SPLIT-ST	820	248	46	9
SPLIT-TR	753	234	44	5
SPLIT-STR	758	239	43	7
#compounds in reference text:	1,105	1,105	396	193

Table 13.3.: **all**: #compounds produced; **ref**: exact reference matches; **new**: unknown to parallel training data; **new***: unknown to target language training data. With **SC** = re-implementation of Stymne and Cancedda (2011), **T**= use full **T**arget feature set, **TR** = use **T**arget features, but only a **R**educed set, **ST** = use **S**ource language features plus **T**, **STR** = use **S**ource language features plus **TR**

source language features is larger for the reduced feature set (SPLIT-TR vs. SPLIT-STR = +0.35) than for the usage of all features (SPLIT-T vs. SPLIT-ST = +0.19) However, the BLEU metric alone does not optimally reflect compound related improvements, as it is dominated by four-gram precision and the length penalty, whereas compound processing leads to improvements at the unigram-level. We thus performed a number of additional manual evaluations, which revealed the improved compound translations of SPLIT-STR when compared to the UNSPLIT system (Section 13.3) and, moreover, presented sentence-level translations of the two systems to external human annotators (Section 13.4).

13.3. A Closer Look at Compounds

Our primary goal is not to improve BLEU scores, but to improve translation quality. As BLEU is not sensitive enough to capture compound-relevant improvements, we report here on a compound-aware manual evaluation: first, we manually screened the German reference set and the translation outputs of all systems to identify compounds. Then, we calculated the number of compounds matching the reference for each experiment. A compound was considered only if it occured in the same sentence of reference and test data. Finally, we also checked whether these had occured in the parallel training data and the big monolingual training data respectively. The results are given in Table 13.3.

It can be seen that the compound processing systems (Split*) not only produce more compounds and more reference-matching compounds, but also a considerable number of compounds that have not occured in the parallel training data. This shows that compound processing in fact enables the production of unseen compounds in English to German Smt. See Table D.1 in Appendix D for a listing of all compounds the system produced, which have not occured in the parallel training data.

Having a closer look, we find that even though Split-Str finds fewer reference matches than for example Split-T or Split-St, it is more precise when producing compounds, as it produces fewer compounds overall. However, comparing the number of compounds our best system[92] Split-Str produced (758) and the number of exact reference matches thereof (239), we find a gap of 519 compounds. We take the English source words of all produced compounds into consideration to assess whether the compounds are adequate translations thereof in Section 13.3.1 (approximating the *precision* of compound processing in Smt). Moreover, we can see from Table 13.3 that only 239 of the 1,105 compounds in the reference have been produced by Split-Str. In Section 13.3.2, we will give details on what Split-Str produced in places where the reference contained a compound, again taking the English input into consideration (approximating *recall*).

13.3.1. Precision

The results in Table 13.3 have shown that the compound processing systems (Split*) generally produce more compounds (758) than Unsplit (661) and Raw (646), and only a small subset thereof matches the German reference exactly (239). However, the German reference translations are only one of many possible ways how an English input sentence can be translated correctly into German. Compounds produced by our systems that do not match the reference are not neccesarily wrong translations. We thus manually investigate the quality of the compounds produced by Split-Str, taking into consideration the English source words from which they have been translated.

A detailed analysis of the 758 compounds that Split-Str produced is given in Table 13.4. For each category, we give an example for Split-Str, the English input words (EN) and the German reference translations (REF), and the respective counts. In terms of precision, it can be seen that most of the compounds our system produced (606 of 758, ∼80%) are correct translations of the English, even though only 239 of them ex-

[92] *“Best”* in terms of yielding the highest Bleu score, cf. Table 13.2.

translation? → **compound? ↓**		**correct translation**	**wrong lexemes**	**wrong merging**	**total counts**
1	**STR:** ✓	Verkehrsunfall	Rentenplänen	Getränkekontrolle	
	EN: ✓	traffic accident	pension plans	alcoholic beverage control mission	
	REF: ✓	Autounfalls	Rentenkasse	Alkoholkontrollkommission	
	counts:	441	45	16	502
2	**STR:** ✓	Lieblingsbuch	Berufsauffassung	*Arbeitsfreitage	
	EN: ✓	favourite book	professional opinion	non-working days	
	REF: ✗	bevorzugtes Buch	offizielles Gutachten	arbeitsfreie Tage	
	counts:	121	17	11	149
3	**STR:** ✓	Abendessen	Prunkstück		
	EN: ✗	dinner	jewel		
	REF: ✓	Festessen	Klangjuwel		
	counts:	18	8	0	26
4	**STR:** ✓	Ehefrau	Platzpech	*Geradenummern	
	EN: ✗	wife	unlucky position	straight numbers	
	REF: ✗	Frau	unglückliche Schlägerposition	nüchterne Zahlen	
	counts:	26	17	38	81
total counts:		606	87	65	758

Table 13.4.: Detailed analysis of the compounds produced by the **Split-STR** system.[93] Unusual German compounds are marked *.

actly matched the reference. A typical example is the translation of "traffic accident" into *"Autounfall"* (lit. = "car|accident") by the human reference translator, which is correct, and into *"Verkehrsunfall"* (lit. = "traffic|accident") by SPLIT-STR, which correct and almost synonymous to *"Autounfall"*. However, these cases are not captured by BLEU scores, as they are calculated string-based. Moreover, there is a fair number of erroneous lexical choices (87 of 758) made by the decoder. The number of erroneous mergings amounts to less than 10% (65 of 758). In addition, the compounds in Table 13.4 were also categorised according to the presence of a compound in the English input and the

[93] Glosses *Verkehrsunfall* = "traffic accident" *Autounfalls* = "car accident", *Rentenplänen* = "pension plans" (in the the literal sense of "plan"), *Rentenkasse* = "pension plans" (in the sense of "plan" = "fund"), *Getränkekontrolle* = "beverage|control", *Alkoholkontrollkommission* = "alcolohlic control commission", *Lieblingsbuch* = "favourite book", *bevorzugtes Buch* = "prefered book", *Berufsauffassung* = "professional view", *offizielles Gutachten* = "official report", *Arbeitsfreitage* = "work|free|days", *arbeitsfreie Tage* = "non-working days", *Abendessen* = "dinner", *Festessen* = "banquet", *Prunkstück* = "showpiece", *Klangjuwel* = "sound jewel", *Ehefrau, Frau* = "wife", *Platzpech* = "position|misfortune", *Schlägerposition* = "racquet position", *Geradenummern* = "straight|numbers" (in the sense of "even"), *nüchterne Zahlen* = "straight numbers" (in the sense of "plain").

group ID	English	Reference	Example	Unsplit	Str
1: shared lexical concept with the reference translation				**331**	**346**
1a: perfect match	inflation rate	Inflationsrate	Inflationsrate	185	239
1b: inflection wrong	raw material preices	Rohstoffpreise	Rohstoffpreisen	40	44
1c: merging wrong	lawyers movement	Anwältebewegung	Anwaltsbewegung	5	9
1d: no merging	police chief	Polizeichef	Polizei Chef	101	54
2: correct translation of the English				**437**	**462**
2a: compound	central banks	Notenbank *note|bank*	Zentralbanken *central|banks*	92	171
2b: no compound	vegetable oils	Speiseöl *food|oil*	pflanzliche Öle *vegetable oils*	345	291
3: wrong translation of the English				**337**	**297**
3a: compound	state budget	Staatshaushalts *state|budget*	Haushaltsdefizite *budget|deficit*	12	42
3b: no compound	spur lines	Nebenlinien *side|line*	Ansporn Linien *motivation|lines*	325	255
Total number of compounds in reference text:				**1,105**	**1,105**

Table 13.5.: Categories for detailed manual compound evaluation, starting from the compounds in the reference text. The counts for **Unsplit** and **Str** are given in the two rightmost columns.

German reference, respectively. In most cases, all system outputs we compared contained a compound (502 of 758). Quite often, the English input and the Split-STR system contained a compound (149 of 758). For example, the Split-STR system produced 121 correct translations of the English input compounds in places where the reference translations did not contain a compound. An example is the English input “favourite book”, which can be correctly translated into either *“bevorzugtes Buch”*, as it happened for the reference translation or into the compound *“Lieblingsbuch”* as it happened for Split-STR. We found only few cases (26 of 758) in which the English input did **not** contain a compound, but both the reference and Split-STR did. Finally, sometimes only Split-STR produced a compound (81 of 758), most of which were either erroneously merged (38 of 81) or contained wrong lexemes (17 of 81).

13.3.2. Recall

We have seen from Table 13.3 that the Smt systems only produced a small subset of the compounds in the reference. However, only exact string matches are counted there, and usually, there is more than one possible way to translate an Enlish input sequence correctly. Here, we start from the 1,105 compounds found in the German reference translation and manually examine the corresponding translations of Unsplit and Split-STR, which performed comparably well in terms of Bleu scores (14.74 vs. 14.61).

Table 13.5 illustrates the different categories according to which the compounds of these two systems were classified, including an example for each category: 1) lexically matching the reference, i.e. using the same lexical concept, even though maybe not exactly matching due to b) an erroneous inflection of the compound or c) a wrong choice of modifier form in the merging process or d) no merging; 2) correct translations of the English input, despite using different lexemes than in the reference translation with a) producing a compound and b) not producing a compound and 3) wrong translations of the English input, either a) with or b) without producing a compound.

Results In the rightmost two columns of Table 13.5, we give the results for Unsplit and Split-STR. Besides the higher number of exact reference matches of Split-STR (239 vs. 185), which was already given in Table 13.3 above, it can be seen from Table 13.5 that Split-STR yields more shared lexical concepts with the reference (346 vs. 331) and more correct translations of the English input (462 vs. 437) than Unsplit. In contrast, Unsplit produced more wrong translations of the English input (337 vs. 297). Two details of Table 13.5 merit further attention: in category 1d), i.e. no merging but same lexical concept as used in the reference translation, we counted only half as much occurences for Split-STR than for Unsplit (54 vs. 101) and in category 2a, we counted almost twice as many cases for Split-STR than for Unsplit (171 vs. 92). Both of these observations confirm that compound processing leads to more compounds in Smt output, which are correct translations of the English input. Recall that all of the mentioned differences[94] in terms of compound translation quality of the two systems are not captured by the previous automatic evaluation using Bleu.

Examples Some examples of the detailed manual analysis are given in Table 13.6: it contains the English input words, the translations of Unsplit and Split-STR, respectively, and the reference translation. It can be seen that for "teddy bear", whose correct German translation *"Teddybären"* is missing in the parallel training data, the Unsplit system produced *"tragen"* ("to bear") instead of *"Bär"* ("bear"). *"Emissionsverringerung"* (cf. Table 13.6) is a typical example of group 2a): a correctly translated compound that does not lexically match the reference, but which is semantically very close to the refer-

[94] Except for the exact reference matches.

English input	**Unsplit**		**Split-Str**		**Reference**
	Compound processing yields better translations				
teddy bear	3b	Teddy tragen *Teddy, to bear*	1a	Teddybären *teddy bear*	ˆTeddybären
emissions reduction	2b	Emissionen Reduzierung *emissions, reducing*	2a	Emissionsverringerung *emission decrease*	Emissionsreduktion
fine	3b	schönen *fine/nice*	2a	Bußgeld *monetary fine*	Geldstrafe
	Compound processing yields equal or worse translations				
table tennis	1d	Tisch Tennis *table, tennis*	3a	Spieltischtennis *play table tennis*	ˆTischtennis
credit-card market	1d	Kreditkarte Markt *credit-card, market*	3a	Kreditmarkt *credit market*	Kreditkartenmarkt
rotation rate	1d	Tempo Rotation *rate, rotation*	3a	Temporotation *rate rotation*	ˆRotationstempo

Table 13.6.: Examples of the detailed manual compound analysis for **Unsplit** and **Str**. Compounds not having occured in the parallel training data are marked ˆ. The categories (e.g. 3b) refer to the ones introduced in Table 13.5 above.

ence. The same applies for *"Bußgeld"*, a synonym of *"Geldstrafe"*, for which the Unsplit baseline selected *"schönen"* ("fine, nice") instead. Consider also the wrong compoundings of Split-STR: *"Tischtennis"* is combined with the verb of the sentence, *"spielen"* ("to play"), into *"Spieltischtennis"*. In contrast, Unsplit leaves the two words separate, which does not hurt understanding here. For *"Kreditmarkt"*, Split-STR dropped the middle part *"Karte"* ("card") and again, Unsplit leaves the correct words separate. An exception is *"Rotationstempo"* (= "rotation rate") which both systems got wrong, as the decoder outputs the two parts in the wrong order. Here, Split-STR produces *"Temporotation"* where the head and modifier of the compound are switched, which hurts the understanding of the translation. However, the current compound processing pipeline does not allow for permutation of the compound parts after decoding.

Generalisation I We investigated the effect of compound splitting and reduction to the underspecified representation in terms of generalisation for the translation of "teddy bear" (which occured in Table 13.6 above). First, we extracted all words containing the substring *"bär"* ("bear") from the original parallel training data (Raw), from the underspecified Unsplit version, and from its underspecified Split version. We found 17 different word types containing "bear" in the Raw and Unsplit versions of the parallel training data, respectively, see Table 13.7 for a detailed overiew of counts.[95] In Split-Str,

[95] Glosses: *Bär/en* = "bear/s", *Bärendienst* = "disservice", *Bärenfarmen* = "bear farms", *Bärenfell* = "bear fur", *Bärengalle* = "bear bile", *Bärenhaut* = "bear skin", *Bärenmarkt* = "bear market", *Braun-*

Raw		Unsplit		Str	
19	Bär	33	Bär<+NN><Masc><Sg>	94	Bär<+NN><Masc><Sg>
26	Bären	12	Bär<+NN><Masc><Pl>	29	Bär<+NN><Masc><Pl>
42	Bärendienst	42	Bär<NN>Dienst<+NN><Masc><Sg>		
1	Bärenfarmen	1	Bär<NN>Farm<+NN><Fem><Pl>		
2	Bärenfell	2	Bär<NN>Fell<+NN><Neut><Sg>		
1	Bärengalle	1	Bär<NN>Galle<+NN><Fem><Sg>		
1	Bärenhaut	1	Bär<NN>Haut<+NN><Fem><Sg>		
1	Bärenmarkt	1	Bär<NN>Markt<+NN><Masc><Sg>		
1	Braunbär	3	braun<ADJ>Bär<+NN><Masc><Sg>		
3	Braunbären	1	braun<ADJ>Bär<+NN><Masc><Pl>		
1	Braunbärengebiete	1	braun<ADJ>Bär<NN>Gebiet<+NN><Neut><Pl>		
1	Braunbär-Population	1	Braunbär–Population<+NN><Fem><Sg>		
18	Eisbären	2	Eis<NN>Bär<+NN><Masc><Sg>		
		16	Eis<NN>Bär<+NN><Masc><Pl>		
2	Eisbärenpopulation	2	Eis<NN>Bär<NN>Population<+NN><Fem><Sg>		
1	Eisbärpopulationen	1	Eis<NN>Bär<NN>Population<+NN><Fem><Pl>		
1	Schwarzbär	2	schwarz<ADJ>Bär<+NN><Masc><Sg>		
1	Schwarzbären				

Table 13.7.: Example of how compound splitting helps to sum up different occurrences of "bears" in simple words, compound heads or modifiers.

all of these "bears" are reduced to 94 occurences of Bär<+NN><Masc><Sg> and 29 occurences of Bär<+NN><Masc><Pl>. These numbers demonstrate that compound processing allows to access all occurrences of the word. This leads to higher frequency counts and thus enhances the probabilities for correct translations.

Generalisation II The "teddy bear" example above showed how frequency counts and probabilities are enhanced through compound splitting. In the following, we give another example confirming this finding. It shows that compound processing allows to form new compounds from words that have previously mostly occured within compounds (as opposed to separate words) in the parallel training data. In contrast to "bear", which occured 45 times as a separate word, even in the original training data (without splitting), "miniature" occured only once as a separate word, but 16 times in compound constructions. An overview of translations and counts from the original parallel training data (without splitting) is given in Table 13.8. At testing time, the compound "miniature camera" is to be translated, whole sentences are given in Figure 13.1. Compared to the Raw and the Unsplit baseline, it can be seen that only the compound processing system Split-STR is able to produce the correct translation *"Miniaturkameras"*.

bär/en = "brown bear/s", *Braunbärengebiete* = "brown bear territory", *Braunbär-Population* = "brown bear population", *Eisbären* = "polar bear/s", *Eisbärenpopulation/en* = "polar bear population/s", *Schwarzbär/en* = "black bear/s".

f	English training data	f	German training data
5	miniature version	2	**Miniatur**version
		2	Kleinformat
		1	**Miniatur**ausgabe
4	Europe in miniature	2	Kleineuropa
		1	Europa im Kleinen
		1	kleines Europa
2	miniature Europe	1	**Miniatur-**Europa
		1	Europa im **Miniatur**format
2	Yugoslavia in miniature	1	Jugoslawien im Kleinformat
		1	Jugoslawien en miniature
1	miniature Americas	1	Miniamerikas
1	miniature national flags	1	**Miniatur**landesfahne
1	miniature United Nations	1	**Miniatur**ausgabe der Vereinten Nationen
1	miniature	1	**Miniatur**

Table 13.8.: Translations of the 17 occurrences of "miniature" found in the parallel training data. Note that it occured only once as a simple word. The German lexeme *"Miniatur"* is used in 8 of the 17 translations.

English input	The images, taken with **miniature cameras** attached to troop helmets, are monitored by a command centre in Kandahar and then transferred to London from where they are uploaded onto the Internet.
German reference	Die mit auf Helmen befestigten **Miniaturkameras** gedrehten Aufnahmen werden am Luftwaffenstützpunkt in Kandahar kontrolliert und anschließend nach London geschickt, wo sie ins Internet gestellt werden.
German RAW	Die Bilder , die mit **kleinen Kameras** an Truppenabzug Helme , verfolgt von einem Kommandozentrale in Kandahar und dann in London , von wo aus sie uploaded auf das Internet .
German Unsplit	Die Bilder , die mit **Miniaturausgabe** der Kameras zur Überwachung der Helme , Truppen durch eine Kommandozentrale in Kandahar und dann nach London verlegt , aus der sie uploaded aufs Internet .
German Split-STR	Die Bilder , die mit **Miniaturkameras** beigefügt Truppenhelmen überwacht werden , von einer Kommandozentrale in Kandahar und dann nach London , von wo aus sie uploaded ins Internet .

Figure 13.1.: Sentence Nr. 84, taken from the WMT 2009 testset, containing "miniature cameras", together with its human reference translation and the translations by the raw baseline, the unsplit baseline, and our best performing compound system STR.

(a) Fluency: without reference sentence

$\kappa = 0.3631$		**person 1**			**all**
		Str	**Unsplit**	**equal**	
person 2	**Str**	24	6	7	37
	Unsplit	5	16	9	30
	equal	6	2	9	17
	all:	35	24	25	84

(b) Adequacy: with reference sentence

$\kappa = 0.4948$		**person 1**			**all**
		Str	**Unsplit**	**equal**	
person 2	**Str**	23	4	5	32
	Unsplit	4	21	7	32
	equal	5	3	12	20
	all:	32	28	24	84

Table 13.9.: Human perception of translation quality.

13.4. Human Perception of Translation Quality

We presented sentences of the Unsplit baseline and of Str in random order to two native speakers of German and asked them to rank the sentences according to preference. In order to prevent them from being biased towards compound-bearing sentences, we asked them to select sentences based on their native intuition, without revealing our focus on compound processing. Sentences were selected based on source language sentence length. Starting from the full set of 1026 sentences, we kept only those with 10-15 words (178 sentences remaining), In these sentences, either the reference or our system had to contain a compound (95 sentences remaining). After removing duplicates, we ended up with 84 sentences to be annotated in two subsequent passes: first without being given the reference sentence (approximating fluency), then with the reference sentence (approximating adequacy). The results are given in Table 13.9. Both annotators preferred more sentences from our system, but the difference is clearer for the fluency task.

13.5. Comparison to Google Translate

We also compared the performance of our compound processing system to a state-of-the art high quality translation system, Google Translate.[96] However, as Google Translate is (probably) trained on more in-house parallel training data than the datasets we used from the 2009 WMT shared task, Bleu scores would not have been directly comparable and have thus not been calculated. Instead, we compared the numbers of the compounds produced by Google Translate with the compounds we manually identified in the German reference test set through exact string matching. We found that Google Translate

[96] `http://translate.google.com` as of October 15th 2013.

produced 237 exact reference matches, whereof 35 compounds[97] were unknown to the parallel data of WMT 2009 and thereof 3 were unknown to the target language data we used[98] (= 3 occurrences of *"MRSA-Infektion"* = "MRSA-infection"). Along the way, we also found some wrongly hyphenated compounds in Google Translate's output that included closed class words (*"und"* in *"Gewerbe-und"*="industry-and") or a mix of German and English words like *"Risk-Studenten"* ("risk-students").

This indicates that Google Translate might use some kind of compound processing of hyphenated words. However, most likely, the 35 new compounds (with respect to the parallel data of the 2009 WMT shared task) that were found in the Google Translate experiment originate from additional parallel training data at the disposal of the Google system, and not neccessarily from extensive compound processing. However, this conclusion is drawn from an observation. We have no knowledge about how compound processing is or is not integrated into Google Translate, as the tool is constantly being improved, our observations might no longer hold in the future.

13.6. Chapter Summary

This chapter dealt with the integration of compound processing in end-to-end English to German Smt. We presented details on the Smt systems we use and described different experiments we performed to examine the effects of compounding and re-inflection on Smt quality: we implemented two contrastive baselines without compound processing and we compared a number of different compound merging strategies in the compound processing systems. The results showed that the positive effects of compound processing on translation quality are not reflected in terms of Bleu scores, where the best system performs slightly (though not significantly) worse than the inflection-handling baseline. In subsequent manual evaluations, however, we could find more correct compounds in the output of the compound processing system, many of which did not occur in the parallel training data.

[97] Namely: *MRSA-Infektion, Antibiotika-resistente, MRSA-Infektion, MRSA-Infektion, MRSA-Infektionen, U-Bahnlinie, Koalitionsmehrheit, Fahreigenschaften, Kolonialarchitektur, Tischtennis, Wohnkomplex, Welttournee, Schneemann, Schneemann, Luxusmarken, Goldfäden, Rosenstrauch, Busladungen, Bildungsexperte, Bundesagentur, Nachbarplaneten, Abendstern, Bandmitglieder, Plattenfirma, Drehbuchautor, Kopfgeldjäger, Wohnnutzung, Holzkiste, Holzkisten, Werkverzeichnis, Museumsdirektor, Holzkisten, Museumsdirektor, Holzkisten, Großaktionär*

[98] These numbers are to be compared to the numbers in Table 13.3 (p. 177) above.

14. Related Work

In this chapter, we review previous approaches to machine translation into morphologically rich languages. This translation direction and all tasks associated with it has recieved much less attention in the past than the opposite one, where any morphologically rich language is (mostly) translated into English (see Chapter 10 for details). One the one hand this might be due to the main interest of the SMT community to translate into English as a pivot language. On the other hand it is obviously easier to delete morphological information not present in the other language than to generate morphologically rich and coherent output.

Nevertheless, there has recently been a growing interest in translating into morphologically rich languages and the available approaches differ with respect to the languages they cover. The remainder of this chapter centres around the two relevant topics for SMT into German: We address compound merging in Section 14.1 and inflectional morphology in Section 14.2. Our main focus is on approaches related to SMT.

14.1. Compound Merging

As we have seen from the previous chapters, compound merging can easily be integrated into SMT as a post-processing step after translation into a split version of the target language. We distinguish three main trends in previous work: list-based approaches, symbolic approaches and CRF-based approaches, all of which we will describe in more detail below.

List-based Approaches Popović et al. (2006) present an intuitive list-based approach to compound merging for English to German SMT. Prior to training, a list of all compounds and their component words is compiled as a by-product of data preparation, which included compound splitting. After translation, only component words that are found on this list are restored. This approach has the advantage of never creating invalid German compounds. However, it is limited to the coverage of the list, and Stymne (2009) found that the list-based merging alone tends to merge too often. She shows that imposing POS-constraints on the list-based merging helps to easily overcome this issue.

Another list-based approach to compound merging is described in Fraser (2009), where the list is not used as an external ressource but realised as a two-step word alignment instead. The idea is to first align from English into a reduced and split German representation and to then perform a second word alignment between the reduced German and the fully specified German.

In Arabic, compounding consists of the combination of prefixes and suffixes, mostly corresponding to prepositions, personal pronouns, or the conjuction "and" in English, with a word stem. Compounding is semi-productive as possible prefixes and suffixes are closed word classes and their number is limited. Badr et al. (2008) investigated a list-based merging approach for Arabic, which is similar to Popović et al. (2006) and compared its performance to a a set of hand-crafted recombination rules. They found that a the rule-based approach clearly outperformed the list-based approach and that a combination of both performed best. El Kholy and Habash (2010) extended the approach of Badr et al. (2008) in that they integrated a 5-gram language model into the process which further improved the performance.

Symbolic Approaches One major drawback of the list-based approaches to compound merging is their limited coverage standing against the unlimited productivity of German compounds. So-called symbolic approaches overcome this shortcoming through using a special markup applied to each former compound part in the course of the splitting process. This distinguishes former modifiers from former heads and simple words. Intuitively, a word that has been found in modifier position of a compound in the training data can take this position again at testing time. Moreover, the markup allows to impose constraints on the merging process.

An extensive overview of different markup schemes and experiments for German is given in Stymne (2009). For compound splitting, she makes use of POS-based compound splitting (see Section 6.3 or Stymne (2008) for details) and uses POS-tags as an additonal factor in SMT[99] to promote the correct order of compound modifiers and translation during decoding. In the following, we give examples for two markup schemes,

[99] See Koehn and Hoang (2007) for an introduction into factored SMT.

that are relevant for compound merging. Consider the compound *"Verkehrsunfall"* (= "traffic accident"). The markup can either be applied to the POS only, adding "-PART" to the former, the lemmatised modifier *[Verkehr NN-PART Unfall NN]*, or to both the words and the POS as in *[Verkehrs# NN-PART Unfall NN]*, where the former modifier is not lemmatised and furthermore clearly marked with "#". Stymne (2009) investigated the impact of these markup schemes in combination with different merging strategies. Examples include list-based and symbol-based approaches. In addition, she investigated different POS-constraints, for example that the POS of the modifier must match with the POS of the potential head word. Stymne (2009) performed experiments both with a factored SMT model (with POS as an additonal factor) and without. Compared with the list-based merging approach alone, all of the merging schemes of Stymne (2009) improved translation quality, some of which significantly outperformed the baseline without compound processing. The experiments with the factored SMT model yielded *overall* higher Bleu scores, but there were more *relative* improvements over the baseline without the factored model. Without the factored model, more, though partly erroneous, compounds were created.

Symbolic approaches to compound merging have also been applied to other languages in the past. Virpioja et al. (2007) split Finnish words into word stems and affixes using an unsupervised morphology, with affixes recieving a simple symbolic markup. Translation happens on the morpheme-level and the merging of morphemes after translation helps to create unseen combinations of stems and suffixes. Similarly, Durgar El-Kahlout and Oflazer (2006) split complex Turkish words into stems and affixes. Due to Turkish vowel-harmony constraints, affixes cannot freely be combined with stems. Durgar El-Kahlout and Oflazer (2006) thus check mergings for morphological soundness using a morphological analyser. deGispert and Mariño (2008) split enclitic pronouns from Spanish verbs and mark them with a symbol in order to be able to merge them back after translation.

CRF-based Approaches Stymne and Cancedda (2011) were the first ones who considered compound merging a sequence labeling problem and used conditional random fields (CRFs) to predict merging points. Compounds were split using the POS-based splitting approach described in (Stymne, 2008). They train CRFs on a noisified version of the split training data with a rich feature collection derived from the target language and applied compound merging on held-out test data, instead of real SMT output. Even though modifiers are not reduced to their base forms (which blocks full generalisation over former compound parts), the approach creates compounds that have not occured in the training data.

Fraser et al. (2012) integrated the approach of Stymne and Cancedda (2011) into an end-to-end SMT task, with the following modifications: they use a rule-based morpholgy for compound splitting (Fritzinger and Fraser, 2010), they included source-language features in Crf training and combined compound processing with inflection prediction. However, while possible compound constituents were predicted by the Crf model, the merging itself was realised using a list-based approach, which is again limited to the coverage of this list. Moreover, compound modifiers were not fully reduced in their approach either. Nevertheless, even Fraser et al. (2012) reported on newly created compounds that had not occured in the parallel training data.

Finally, in Cap et al. (2014a) compound merging was taken one step further in that modifiers were reduced to their base forms (allowing for full generalisation over former compound parts) and compounds were generated using a rule-based morphological analyser. More detailed results can be found in Section 13.2 above.

Other Approaches An early approach to compound merging for German in the context of an automatic speech recognition system has been reported by Larson et al. (2000), where compounds are split based on character-ngram statistics. For compound merging, bi- and tri-gram frequencies are collected and merged above certain thresholds. The optimal granularity for merging is found by calculating the language model perplexity of a held-out test set.

Baroni et al. (2002) integrated compound merging into a word prediction system for German. Merging decisions are derived from weighted probabilities of different frequency-based features, which partly overlap with the features Stymne and Cancedda (2011) used, e.g. how often a word occured in modifier vs. head position or as a simple word.

While the approaches reported in this section deal with target-language compound merging, there has also been some work on source language word packing for translations from non-compounding into compounding languages, e.g. by Ma et al. (2007). See Section 10.4 for more details.

14.2. Inflectional Morphology

Compound processing alone is not sufficient for English to German SMT, where the target language exhibits a rich inflectional morphology. Similar to compound processing, inflection handling is usually realised as a two-step process: translate into a stemmed version of the target language and then re-inflect it to fully specified German. In this section, we summarise previous work on stemming and re-inflection for SMT.

14.2.1. Stemming

Here, we summarise previous approaches to stemming in SMT that have been designed to improve translation systems with the morphologically rich language being the source language. They differ with respect to i) the way the morphologically rich language is reduced (i.e. lemmatised, stemmed or simply truncated), ii) being integrated to end-to-end SMT or word alignment alone.

Stemming Procedure On the one hand, the stemming (or lemmatisation) itself can be performed in a supervised manner, using a rule-based morphological analyser (e.g. Nießen and Ney, 2004; Corston-Oliver and Gamon, 2004; Hardmeier et al., 2010), a treebank (Goldwater and McClosky, 2005) or a POS-tagger with a rich morphological annotation scheme (Čmejrek et al., 2003). On the other hand, there have also been unsupervised approaches to stemming in SMT, for example a simple truncation of word endings as in Fraser (2009). Goldwater and McClosky (2005) compared the effects of supervised and unsupervised stemming on Czech to English SMT. In an additional variant, they stemmed only low-frequent words, which slightly improved the performance. This is in line with experiments of Dejean et al. (2003) on a French to English SMT task, who found that selective lemmatisation of low-frequent words performs better than lemmatisation of all words. Nießen and Ney (2000) present a different kind of selective lemmatisation: they lemmatise only words that have not occured in the parallel training data.

Level of Granularity However, stemming is not always benefical for translation performance: removing morphological distinctions which exist in both source and target language sometimes hurts translation quality. An early approach by Al-Onaizan et al. (1999) for translation from Czech to English introduced artificial Czech words in places where Czech uses affixiation to express concepts for which English uses separate words. Popović and Ney (2004) reduced the verbal morphology of Spanish and Catalan verbs into stems and a restricted number of POS tags (e.g. person, conditional tense) that are also distinguished in English.

More recently, Talbot and Osborne (2006) presented a language-independent approach for clustering morphological variants in the source language that are not distinguished in the target language. They investigated the language pairs Czech, Welsh, and French to English and found the largest improvements for the morphologically richer languages Czech and Welsh.

Other approaches use two-pass word alignments to find an optimal level of morphological reduction for the source language. This method is suitable for highly inflected languages using affixiations that correspond to separate words in the target language. Examples include DeNeefe et al. (2008) for Arabic and Bai et al. (2008) for Chinese. More details can be found in Section 10.1, where we described alignment-based approaches in the context of previous work on compound splitting.

Application Range Instead of using the stemmed representation in SMT model training, Talbot and Osborne (2006) use it to improve word alignment and then train the translation models on the original version of the data. This is similar to Popović et al. (2006), who split compounds for word alignment, but then trained the models on unsplit text, after re-adjusting the split alignments to it. Corston-Oliver and Gamon (2004) investigated the effects of lemmatisation on word alignment quality alone, without integration into end-to-end SMT. For translation into a morphologically rich language, Toutanova et al. (2008) present a method where word alignment is trained on fully inflected text, whereas the translation model is estimated on a stemmed variant of the original text and later re-inflected using feature-rich probabilistic models.

Yang and Kirchhoff (2006) proposed hierarchical back off models that, whenever faced with an unknown word, back off to a more general level of morphological description. Similarly, Hardmeier et al. (2010) perform compound splitting and lemmatisation and use a lattice-based approach that allows the decoder to choose among lemmatised and fully specified versions of the current word.

14.2.2. Re-Inflection

Previous approaches to re-inflection in the context of statistical machine translation can be divided into two main groups: factored approaches and approaches to morphological generation.

Factored Approach The factored approach introduced by Koehn and Hoang (2007) allows use morphological features during the translation process. Each level of morphological description (e.g. lemma or POS of the word) is encoded in a factor assigned to the word. For translation from English to German, the usage of morphological features lead to a considerable improvement of noun phrase agreement.

In Avramidis and Koehn (2008) the syntactic structure of the source language (English) is used to enrich the source language representation with morphological features that are marked in the target language (Greek). A factored approach is then trained on the preprocessed source language data and the original target language data. At translation time, not all of the factores must always be used.

Synthetic Phrases More recently, Chahuneau et al. (2013) introduced synthetic phrases, which are used to enrich the original phrase table. Their approach allows to generate unseen inflectional variants of words that have occurred in the parallal data. It requires training two translation models: one on the original data and one on a stemmed version of the target language. The phrases of the latter one are inflected based on the context of the corresponding source language phrase and then added as synthetic phrases to the original phrase table. The final translation model has then access to all phrases, the original and the synthetic ones. The approach is conceptually language-independent; Chahuneau et al. (2013) report on experiments for the translation pairs English to Russian, Hebrew and Swahili.

Morphological Generation The inflection prediction procedure we use in this thesis has been described and implemented by Weller (2009) and Fraser et al. (2012). The basic concept of a two-step translation approach, namely to first translate into a stemmed, underspecified representation of German and to later predict morphological features to generate fully inflected German is based on work of Toutanova et al. (2008). They investigated morphological generation within different end-to-end SMT systems for translation from English into Russian and Arabic. The target language morphological features we use for inflection prediction are similar to those of Toutanova et al. (2008). However, they additionally made use of features derived from the source language input,[100] see Minkov et al. (2007) for a detailed description. Besides the two-step translation approach we adapted, Toutanova et al. (2008) also performed experiments where the English input was directly translated into fully inflected Russian and Arabic, respectively. Then, the SMT output was stemmed and re-inflected using the same models as in the two-step approach. In Arabic morphology, articles, prepositions, personal pronouns and conjunctions are realised as affixes to the word stem. Toutanova et al. (2008) segmented these affixes from the stems (using a rule-based morphological analyser) to get stemmed Arabic training data. In addition to these morphological features, Arabic also allows for case variations. However, these are not marked in the Arabic standard script but expressed using diacritics instead. Habash et al. (2007) investigates the predictability of the Arabic case feature using machine learning on treebank data.

Toutanova and Suzuki (2007) perform case prediction in English to Japanese SMT using n-best list re-ranking techniques. The translation model is trained on stemmed data and during tuning, a number of inflection variants is generated for each of the n-best list entries. These inflections are then weighed against a reference translation. More recently, Clifton and Sarkar (2011) perform morpheme-based English to Finnish SMT. The Finnish training data is decomposed using an unsupervised morphology. Morpheme recombination is integrated into the translation model and afterwards, vowel harmony constraints are implemented using a bigram language model.

While the apporaches mentioned above focused on nominal morphology, deGispert and Mariño (2008) deal with Spanish verbal morphology in English to Spanish SMT. In their approach, Spanish enclitic pronouns are separated from the stems prior to training and re-attached after translation. In contrast, Ueffing and Ney (2003) merge English pronouns with their corresponding stems in order to account for this asymmetry between English and Spanish.

[100] In the current setting, we use source language features only for the compound merging decision, not for inflection prediction.

Part IV.

The Bottom Line

15. Conclusion

The productive compounding of German usually leads classical SMT systems into data sparsity issues. As a consequence of the variety and productivity of the compounds, many words remain untranslated because they have not (or not sufficiently often) occurred in the parallel training data. In contrast, most of the component words of these compounds have in fact occurred in the data. Compound processing for SMT aims to make the translations of the component words accessible to the translation model and thus make the compounds translatable part-by-part.

In the course of this thesis, we investigated whether the usage of a rule-based morphological analyser improves compound processing in Smt for the language pair of English and German. We processed compounds as part of a pre- and postprocessing procedure to a standard Smt pipeline. The comparison to previous approaches showed that morphological analysis leads to improved translation performances for both directions of the language pair. This can be attributed to more precise compound splitting and merging which we showed on held out clean data sets.

Structure The remainder of this chapter is structured as follows: we first summarise our contributions in Section 15.1. Then, we discuss limitations and possible directions for future work in Section 15.2.

15.1. Summary of Contributions

We designed a morphologically-aware compound processing procedure to improve Smt and applied it to both translation directions of the language pair English and German. In the following, we re-visit and our contributions, first for Smt translation from German to English (Section 15.1.1) and then for the opposite direction, from English to German (Section 15.1.2).

15.1.1. German to English

In recent years, it has become common practise to split compounds for German to English SMT. However, most of the systems that have emerged rely on frequency statistics of substrings (Koehn and Knight, 2003) instead of incorporating linguistic knowledge. In this thesis, we compared the performance of our morphologically motivated compound splitting to two widely used, yet linguistically less informed approaches: i) a frequency-based approach of Koehn and Knight (2003) and ii) a POS-based approach inspired by Stymne (2008) and found our approach improving over both of them.

Translation Results We could show that the integration of our compound splitting procedure into a standard SMT system leads to **improved translation performance** in terms of a standard automatic evaluation metric (BLEU, Papineni et al. (2002)). This improvement is **statistically significant** with respect to i) an uniformed baseline without compound splitting and ii) a system incorporating the purely frequency-based approach to compound splitting.

The reduction of unknown words (i.e. words not having occurred in the parallel training data) in the translation testset is a crucial factor for the improvement of the translation, as unknown words cannot be translated by a standard SMT system without compound processing. We could show that our morphologically aware compound splitting leads to the **lowest number of unknown words**, compared to the two previous approaches.

Splitting Accurracy Moreover, we also performed several gold standard evaluations in order to measure compound splitting accurracies independent of their effect on translation quality. Even here, we found **substantial improvements** of our morphologically aware compound splitting when compared to other approaches.

As a by-product of this work, we will make the **gold standards** we created **publicly available**. This makes the results of our compound splitting procedure comparable to future works.

15.1.2. English to German

In contrast to the opposite translation direction, compound processing for translation from English to German has not yet recieved much attention from the SMT community. Besides the problems caused by productive compounding, this translation direction suffers from additional data sparsity due to the rich inflectional morphology of German. It is thus desirable to combine compound processing with inflection handling. Using a rule-based morphological analyser allows to handle both of these phenomena within one joint step. Prior to SMT training, compounds are split and lemmatised, and after translation the rule-based morphological analyser is used to (re-)merge compounds and generate fully inflected word forms.

Translation Results We combined our morphological compound processing with an available inflection handling component and integrated it into a standard English to German SMT system. We found a statistically significant **improvement in translation performance** in terms of a standard automatic evaluation metric when compared to an uniformed baseline without compound processing and inflection handling.

The results in terms of automatically measurable translation performance were less decisive when comparing our system to a baseline which incorporates inflection handling alone, i.e. without compound processing. Here, an additional **human evaluation** revealed that the results of our system were preferred. Moreover, having a closer look at the translations of the compounds in the testset for both of the systems, we found that our system not only produced more compounds than the contrastive system, but also **more correct compounds**. The fact that many of these compounds have not occurred in the parallel training data shows that our approach successfully generates new compounds, if necessary. Working on a **lemmatised representation**, further promotes the **generalisation** over former compound parts and separate occurrences of simple words.

Merging Accurracy The decision which simple words to merge into compounds can be modeled based on a number of different features. We measured the **accurracies** of different feature combinations with respect to a **clean data experiment** of compounds to be re-merged. We found that using **features** derived from the **source language context** improves the prediction accurracy.

Overall Contributions Finally, our successfull application of a rule-based morphological analyser for compound processing in SMT shows how **linguistic knowledge can be used to improve SMT**. Considering this success from the opposite perspective, we find that the fact that the morphological analyser improves SMT performance can be considered a successful **extrinsic evaluation** of the morphological analyser.

15.2. Limitations and Future Work

In this section, we discuss the limitations of our work and give some directions for future work. We divide the remainder of this section into two parts. The first one deals with shortcomings and possible extensions to our approach which can be addressed without requiring major changes in our general processing pipeline (Section 15.2.1). In the second part of this section, we adress additional shortcomings and some more sophisticated possible adaptations of our approach (Section 15.2.2).

15.2.1. Taking it One Step Further

Compositionality The compound processing approch we presented in this thesis is most feasible for compositional compounds, whose meaning can be derived from the meaning of their component words. Our approach relies on the assumption that compositional compounds can be translated part-by-part. Splitting compounds thus gives the translation model access to the translations of their component words. While most German compounds indeed have a compositional semantics and should thus be split prior to translation, an erroneous splitting of non-compositional compounds may lead to translation errors. For example, splitting the non-compositional compound *"Trantüte"* (= "s.o. being lame") into its component words *"Tran|Tüte"* and translating these words would result in something like "fish oil|bag", which is a very misleading translation.

We do not yet explicitly adress non-compositional compounds in our approach. However, non-compositional compounds are often lexicalised and we do not split compounds which are listed in the lexicon of the rule-based morphology. For merging, we use a feature which indicates that one English word has been translated into two (or more) German words. This can be considered to be an indicator for a German compound that has been erroneously split prior to the SMT training and translation.

In the future, we plan to address compositionality in the course of the compound splitting process and leave non-compositional compounds unsplit. One possibility of doing so is to use distributional semantics to approximate the compositionality of German compounds. The underlying idea is to compare the contexts in which a compound occurred with the contexts in which the compound modifier(s) and head occurred, respectively. The more similar these contexts turn out to be, the more likely the compound is compositional. In order to approximate the semantics of the compounds properly, we may use the full analysis depth of SMOR. This means that even lexicalised compounds are split for the calculation of a compositionality score. We have already begun with initial experiments towards this direction, see Weller et al. (2014) for some first results.

Splitting Restrictions Most German compounds are noun compounds, but German also allows for productive adjective compounds and less productive verbal compounds. In contrast to other approaches (e.g. Weller and Heid (2012)), we do not restrict the POS of the compounds to be split. Instead, we trust the morphological analyser and split whatever the analyser allows to be split. Moreover, we also split particles from verbs, in cases where they can occur separated from the verb. In the course of the manual evaluation of compound translations we performed for the English to German translation direction, we restricted ourselves to the translation of compounds and excluded particle verbs. While the decision for compound merging is based on CRF-based predictions, the merging of particles and verbs is performed using POS-based rules. This was one of the reasons for which we did not explicitly evaluate the translations of particle verbs. Another one was to perform the manual evaluation for a manageable subset of the data.

In the future, we plan to perform additional manual evaluations focussing on the translation of particle verbs. Moreover, we will perform separate experiments on splitting compounds and separating particle verbs in order to investigate the impact of each of these components. Future work on particle verbs may be combined with our plans to address compositionality in the compound splitting process. Not all German particle verbs have corresponding English counterparts. The usage of distributional semantics may improve the accurracy of particle verb separation and, as a consequence, also improve translation quality.

Interaction with Inflection For translation from English to German, we currently combine our compound processing approach with an already existing inflection prediction component. Even though the morphological generation with the rule-based morphology happens in one joint step, the two underlying predictions happen separately. We first decide on words possibly being merged into compounds and then, independently of this first decision, decide on their inflection.

In the future, we will investigate whether these two decisions may be modeled within one joint CRF model. Such a model will predict the usual morphological features which previously were modeled in the CRF model for inflection. In addition, it will predict one feature that indicates whether the present word is a compound modifier (and thus remains uninflected).

Syntax-based SMT In this thesis, we integrated our compound processing approach into a standard phrase-based SMT system. The phrases of such systems do not neccessarily coincide with syntactic phrases. When translating from English to German, we have thus no information about the syntactic role of two German nouns occuring next to each other. In the present system, we use CRF-models to predict possible compound mergings. Some of the features we use in these models are based on the syntactic context of the English source sentence. However, we do not have access to the syntax of the German translation output.

In the future, we will integrate our compound processing system into an existing tree-to-tree SMT system. From a compound processing perspective, this system architecture has the advantage that we only have to model compound splitting. Possible compound mergings will be indicated by the syntactic structure of the target sentence. The surface form of the compound (including eventual filler letters etc.) will be generated as usual, using the rule-based morphology. We recently took some first steps towards this direction, too. We integrated not only compound processing but also lemmatisation and re-inflection into an existing tree-to-tree SMT system. First results we obtained were not yet decisive, see Quernheim and Cap (2014) for details. Nevertheless we plan to further investigate this idea in the future, possibly with a string-to-tree SMT system.

Evaluation In terms of evaluation, the primary focus of this thesis was on the accurracy of compound processing on a clean data task and on translation of German compounds. We found that compound processing lead to improved translation performances for both translation directions. Beyond the standard evaluation metrics, we calculated the number of unknown words, i.e. words not having occurred in the parallel training data which are thus not translatable, in the test set. We found that compound processing substantially decreases the number of unknown words, which certainly has a positive impact on the overall translation quality.

However, we have not yet systematically investigated the effect of compound merging on the fluency of the output at sentence level. Moreover, we never separately investigated the (presumably positive) effect of compound processing on single parts of the translation process. For example, it is intuitively clear that compound processing must lead to improved word alignments: more 1:1 alignments are created and the frequency counts of simple words which previously have occurred within compounds are higher. Another example is language modelling. In the English to German translation direction, we use a language model which is trained on the split and lemmatised version of German. We have shown that compound splitting decreases the vocabulary size of the training data. This should have a positive effect on language modelling.

In the future, we will evaluate the impact of compound processing on single components of the translation system in more detail. This will help us understand where exactly our current approach requires further improvements.

15.2.2. Taking it More Steps Further

Pre-/Postprocessing Approach The compound processing approach we presented in this thesis is integrated into a standard SMT pipeline through extensive pre- and postprocessing of the data on which the SMT model is trained. This way, the final SMT model does not interact with the compound processing procedure. For example, the model has no access to different splitting options of a compound in the course of the translation process. Furthermore, for the English to German translation direction, our current approach translates lemmas which are re-inflected in a post-processing step. This procedure might introduce erroneous inflections in cases where the original data contained enough fully inflected instances to provide a correct translation.

In the future, we may follow Dyer (2009) and extend our approach to use lattices instead of single best splits of words. A lattice-based approach could not only be used in the German to English translation direction for compound splitting, but also providing multiple merging options for the opposite translation direction.

As for the combination of compound processing with inflection handling, it may be interesting to follow (Chahuneau et al., 2013). They used synthetic phrases to model unseen inflectional (and derivational) variants of Russian, Hebrew and Swahili. Their approach works as follows: first, a translation model is trained on the original data. Then, a second translation model is trained on a lemmatised (and possibly split) representation of the German training data. For these lemmatised phrases, inflections are predicted (and compounds are merged), depending on the context in which the source language phrase occurred. The resulting phrases are called synthetic phrases. Finally, the original phrase table (which was trained on the original data), is enriched with these synthetic phrases. At translation time, the translation model has access to all phrases.

Portability The fact that we are using a rule-based morphological analyser for compound processing is a strength and at the same time a weakness. In the thesis, we already discussed the strengths (e.g. more accurate splittings, ability to handle compounding and inflection in one joint step) of our approach. One of its major weaknesses is its limited portability. Our compound processing procedure can easily be applied in SMT with German as one of the languages of the translation pair and a language other than English for the other. However, if it is to be applied to a compounding language other than German, major adaptations of the approach are required. First of all, a morphological analyser is required which has similar capacities and lexical coverage for the language it has been conceived for, as the analyser we are using for German in our work. Provided such an analyser, all pre- and postprocessing scripts have to be adapted to the requirements of the language (i.e. to optimally fit to its compounding phenomena) and of the morphological analyser.

In the future, we may train a semi-supervised (or even completely unsupervised) morphology system for German and compare its performance to our original approach (using a rule-based morphological analyser). For the semi-supervised approach, we expect a loss in performance in comparison to the rule-based morphology. If however, the usage of a semi-supervised morphology for compound processing in SMT still outperforms previous approaches, we may apply it to SMT systems for other morphologically rich languages.

Domain Adaptation Besides moving to a different language, one could also think of moving from general German language, to a specialised German language, e.g. for technical domains or the medical domain. Specialised languages usually contain more compounds overall and also more n-ary compounds with n>2. We thus assume a greater positive impact of compound processing as compared to the effects it had on general language. In general, our approach should work as it is. However, depending on the domain, it is may be useful to extend the lexicon of the rule-based morphological analyser with domain-specific terminology.

In the future, we may apply our compound processing procedure to SMT from and to English and German for a specialised domain. If required, we may extend the lexicon with specialised terminology.

Language Modelling When translating from English to German, we use a language model trained on the split and lemmatised version of the German training data. On the one hand, due to a decreased training vocabulary through splitting compounds this should have a positive effect on language modelling performance. On the other hand, we loose n-gram context which might hurt language modelling performance. For German, we usually train 5-gram language models. Consider for example a sequence of 5 words, where the first and the third word is a compound and the others are simple words: *compound word compound word word*. The default language model trained on the original data stores this whole 5-gram sequence of words. However, in the split representation, the same sequence extends to 7 words: *modifier head word modifier head word word*. A 5-gram language model would in this case loose the context of the two last words, when trained on the split representation.

In the future, we may perform experiments with higher order language models. Another possibility to overcome the n-gram related context-loss problem is to use different levels of representation. For example, the language model may be trained with the usual n-gram order but instead using the split version of the data, we could use a modified version of the original data, where each compound is replaced by its head word, see e.g. Botha et al. (2012).

Part V.

Appendix

A. Informal Comparison of Analysis Formats from GerTWOL and SMOR

In this chapter, we briefly show a few examples to illustrate the similarities and differences between the analyses of Gertwol and Smor. Table A.1 gives an overview of the phenomena we compared.

Insertions, (ins.) e.g. *Tag+Zeitung* = *Tag**es**zeitung*. Also referred to as *Fugenelemente*, are handled equally well in both systems: Gertwol highlights the *Fugenelement* by a preceding backslash, whereas Smor drops the *Fugenelement*. For word splitting, it is only important to find the correct split point(s) and the correct form of the words next to the split point(s). If *Fugenelemente* are highligted or not is not relevant.

Umlautung, (uml.) e.g. *Haus (sg)* → *Häuser (pl)*. The phenomenon of *Umlautung*, e.g. in the case of plural formation, works in both systems. Note that even Gertwol restores the original form without *Umlaut* in its analysis.

Insertions with Umlautung, (iwu) e.g. *V**o**lk+Mord* = *V**ö**lk**er**mord*. Sometimes, the insertion comes with an *Umlaut* transformation. Even though Gertwol is able to handle *Umlaut* (see previous paragraph), here, it only indicates the split point of the word, without restoring the original word (cf. also paragraph on insertions above).

Deletions, (del.) e.g. *Kirch**e**+Turm* = *Kirchturm*. In Gertwol, the deleted letter(s) are not displayed in the analysis, only the split point is indicated. In contrast, Smor maps the two words onto their original form.

type	word	GERTWOL	SMOR
insertions	Tageszeitung	"*tag\es#zeitung" S FEM SG NOM]	Tag<NN>Zeitung<+NN><Fem><Nom><Sg>
umlautung	Häuser	"*haus" S NEUTR PL NOM	Haus<+NN><Neut><Nom><Pl>
insertions with umlautung	Völkermord	"*völker#mord" S MASK SG NOM "*völk\er#mord" S MASK SG NOM	Volk<NN>Mord<+NN><Masc><Nom><Sg>
deletions	Kirchturm	"*kirch#turm" S MASK SG NOM	Kirche<NN>Turm<+NN><Masc><Nom><Sg>
transformations	Studiengebühr	*studi\en#gebühr" S FEM SG NOM	Studium<NN>Gebühr<+NN><Fem><Nom><Sg> Studie<NN>Gebühr<+NN><Fem><Nom><Pl>
unknowns	Rapallo-Vertrag	<*rapallo-*vertrag>	Rapallo-<TRUNC>Vertrag<+NN><Masc><Nom><Sg>
ambiguity	Staubecken	"*staub#eck~e" S FEM PL NOM "*stau#becken" S NEUTR SG NOM "*staub#eck" S NEUTR PL DAT	Staub<NN>Ecke<+NN><Fem><Nom><Pl> Stau<NN>Becken<+NN><Neut><Nom><Sg> Staub<NN>Eck<+NN><Neut><Dat><Pl>
	Kontrollausschuss	"*kontroll#aus|schuß" S MASK SG NOM "*kontroll#laus#schuß" S MASK SG NOM	Kontrolle<NN>Ausschuss<NO><+NN>[2] Kontrolle<NN><OO>Laus<NN>Schuss<NO><+NN>
analysis depth	Ausschuss	"*aus|schuß" S MASK SG NOM	Ausschuss<OO><+NN> aus<VPART>schießen<V><SUFF><OO><+NN>

Table A.1.: Comparison of analysis from GerTWOL and SMOR across different compounding phenomena.

Transformations, (tra.) e.g. *Studi**um**+Gebühr = Studi**en**gebühr*. As for deletions, GERTWOL only indicates the split point withour re-transforming the words to their original form, which SMOR does, even in the present example, where it is not clear whether *Studiengebühren* is the result of compounding *Studium* or *Studie* with *Gebühren*.

Unknowns, (unk.) e.g. *Rapallo-Vertrag**, *Eropa-Vertrag***. If at least the last part of the a hyphenated word is known to SMOR, it is correctly analysed. This case frequently appears for proper nouns(*), type-Os(**) or foreign language material. GERTWOL on the other hand, is only able to analyse a hyphenated word if all parts of the word ar known.

Ambiguities, (amb.) e.g. *Kontrollausschuss = Kontrolle+ Ausschuss || Kontrolle + Laus + Schuss* or *Abteilungen = Ab+teil+ungen, Abtei+Lungen, Abt+Ei+Lungen* or *Staubecken = Staub+Ecken || Stau+Becken*. Some words have more than one possible splitting. As can be seen from the examples table[2], both systems are able to identify such ambiguities.

Analysis depth, (and.) e.g. *Ausschuß vs aus/schießen*. In many cases SMOR's analysis goes deeper than GERTWOL's in that it does not only find the split point, but also retransforms the word parts to their original form (as listed in a lexicon), even if the last part of a word is a verb participle.

Orthography, (ort.) e.g. *Kontrollausschu**ß** vs. Kontroll**l**ausschu**ss***. A short note on orthography: GERTWOL covers only the old German orthography convention, whereas SMOR covers both, the old and the new and even displays to which the acutal word belongs to (<OO> = old/<NO> = new orthography). In the example "Kontrollausschuß" (<OO>), the wrong analysis with "Laus" would not appear in new orthography, as there, 3 identical letters in a row are allowed, i.e. only the word *Kontrolllausschuss* would be uniquely analysed as "Kontrolle|Laus|Schuss".

[2] <Masc><Nom><Sg> are left out from the table for readability reasons.

B. Gold Standard Evaluation of Compound Splitting: Additional Results

B.1. Translational Correspondences Gold Standard

part size → ↓ part frequency		3 characters	4 characters	5 characters	6 characters
1	precision	14.26	22.76	30.80	37.24
	recall	51.95	54.55	52.60	35.06
	accuracy	90.12	94.00	95.66	96.72
3	precision	14.92	23.61	32.03	37.59
	recall	52.60	55.19	52.60	34.42
	accuracy	90.48	94.18	95.76	96.74
5	precision	14.82	23.45	32.06	37.41
	recall	51.30	53.90	51.95	33.77
	accuracy	90.58	94.18	95.74	96.72
10	precision	15.12	23.84	32.64	37.78
	recall	50.65	53.25	51.30	33.12
	accuracy	90.86	94.23	95.84	96.74
25	precision	15.64	24.32	32.91	38.28
	recall	49.35	52.60	50.00	31.82
	accuracy	91.30	94.42	95.88	96.76
50	precision	18.14	27.33	32.44	38.33
	recall	54.55	57.14	47.40	29.87
	accuracy	91.84	94.70	95.90	96.78

Table B.1.: Exploration of different settings of the **extended frequency-based splitting** approach wrt to the **translational correspondences** gold standard.

B.2. Linguistic Gold Standard

part size →		3 characters	4 characters	5 characters	6 characters
↓ part frequency					
1	precision	34.62	46.23	55.72	60.00
	recall	60.66	58.01	46.99	31.15
	accuracy	78.03	84.35	86.89	86.60
3	precision	35.23	46.22	56.26	60.21
	recall	60.29	56.69	46.63	30.87
	accuracy	78.47	84.16	86.97	86.58
5	precision	35.71	47.38	57.08	60.64
	recall	60.29	57.74	46.63	30.87
	accuracy	78.84	84.69	87.07	86.63
10	precision	36.49	48.09	57.53	60.65
	recall	59.93	57.29	46.27	30.60
	accuracy	79.46	84.98	87.10	86.60
25	precision	37.98	49.92	59.18	61.71
	recall	59.29	56.56	45.81	30.24
	accuracy	80.48	85.63	87.30	86.60
50	precision	38.83	51.01	60.32	63.56
	recall	57.92	55.37	44.99	29.78
	accuracy	80.99	85.84	87.28	86.60

Table B.2.: Exploration of different settings of the **extended frequency-based splitting** approach wrt to the **linguistic** gold standard.

C. SMT Evaluation of Compound Splitting: Additional Results

In Table C.1, we give translations of sentence Nr. 31 of the wmt 2009 test set, where the German input mostly differs in the splitting of *"Warenhäusern"* (= "warehouses")[101]. Ideally, the word should be left unsplit, as happened in the Baseline and the two *SMOR -d* experiments. Even though it occurred in the parallel data, it is left untranslated in the baseline, probably due to the low frequency of the plural accusative form. Note also that the (theoretically) correct splitting into *"Ware"* (= "ware") and *"Häusern"* (= "houses") leads to a correct translation for *SMOR*, but to a unusual translation for *POS*. The erroneous splitting into *"waren"* (= "were") and *"Häusern"* (= "houses") lead to a correct translation for *basic freq*, but to a too literal and wrong translation for *extended freq.*. Recall from Section 4 that translations obtained through SMT depend on the interaction of numerous different components, which makes translations hard to track.

In Table C.2, we give translations of sentence Nr. 328 of the wmt 2009 test set, in which the German input mostly differs in the splitting of *"Seitenhalt"* (= "lateral support"). Ideally, this compound should be split into *"Seite"* (= "side") and *"Halt"* (= "support"). As can be seen, this correct splitting is achieved by *POS* and all three *SMOR* approaches. While the baseline outputs the word untranslated as is, neither of the other systems outputs an acceptable translation either. Most outputs contain at least *"side"*, except for the *extended frequency based* system, where the erroneous splitting into *"seit|Halt"* (= "since|support") leads to the disturbing translation "since cohesion" and the output of the *SMOR -d* system, whose translation of the correct splitting *"Seite|Halt"* (= "side|support") into "hand has confined" is even more puzzling to understand. A closer look a the word alignment of the sentences revealed that *"Seite"* was aligned with "hand" (probably originating from the expression "on the other hand" - *"auf der anderen Seite"*)

[101] Note that we give these examples properly cased here, as this enhances readability of German. However, as is common practice in SMT, the input files are lowercased prior to translation.

Reference	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **warenhäusern** gar nicht mehr auf die preise schaut .
	We found that most shoppers in the three **outlets** visited were not really interested in prices .
Baseline	our experience shows that the majority of customers in the three **warenhäusern** no longer look at the prices .
basic freq.	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **waren häusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **stores** no longer on prices .
extended freq.	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **waren häusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **were houses** no longer on prices .
POS	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **ware häusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **commodity houses** no longer look at the prices .
SMOR	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **ware häusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **department stores** no longer on prices .
SMOR -d	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **warenhäusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **department stores** no longer on prices .
SMOR -d NN	unsere erfahrung zeigt , dass die mehrheit der kunden in den drei **warenhäusern** gar nicht mehr auf die preise schaut .
	our experience shows that the majority of customers in the three **department stores** no longer on prices look .

Table C.1.: Translations of sentence Nr. 31 of the wmt 2009 test set

Reference	auch der **seitenhalt** hat sich gegenüber dem 206 wesentlich verbessert .
	Also , the **lateral seat support** has improved markedly in comparison with the 206 model .
Baseline	also the **seitenhalt** has substantially improved the 206 .
basic freq.	auch der **seiten halt** hat sich gegen über dem 206 wesentlich verbessert . also , the **sides** against the 206 considerably .
extended freq	auch der **seit halt** hat sich gegen über dem 206 wesentlich verbessert . *also the **since support** has against over the 206 considerably improved .*
	The **since cohesion** has 206 considerably .
POS	auch der **seite halt** hat sich gegenüber dem 206 wesentlich verbessert . the **side** has 206 improved substantially .
SMOR	auch der **seite halt** hat sich gegenüber dem 206 wesentlich verbessert . the **side** has 206 considerably .
SMOR -d	auch der **seite halt** hat sich gegenüber dem 206 wesentlich verbessert . *also the **side support** has in comparsion with the 206 considerably improved.*
	the **hand** has **confined** itself to the 206 considerably .
SMOR -d NN	auch der **seite halt** hat sich gegenüber dem 206 wesentlich verbessert . the **side** has 206 considerably .

Table C.2.: Translations of sentence Nr. 328 of the wmt 2009 test set

and *"Halt"* was aligned to "confined".

These examples show that a correct splitting does not neccessarily leads to a correct translation. Nevertheless, correct splittings are still good prerequisites for better translations, as the improved translation quality scores showed.

D. SMT Evaluation of Compound Merging: Additional Results

compound	**Sc**	**T**	**St**	**Tr**	**Str**
Miniaturkameras	X	X	X	X	X
Schießspiel	X	X	X	X	X
Wirtschaftsentwicklungszentrum	X	X	X	X	X
Streikausschuss					X
Streikausschüsse	X	X	X	X	X
Garagenrock	X	X	X	X	X
Koalitionsmehrheit	X	X	X	X	X
Kultsymbolen	X	X	X	X	X
Kultautos	X	X	X	X	X
Kolonialarchitektur	X	X	X	X	X
Tischtennis			X		
Teddybären	X	X	X	X	X
Teddybären	X	X	X	X	X
Unternehmensprofils			X		
Klimapropaganda	X	X	X	X	
Fahrzeugmodell	X	X	X	X	X
Gesundheitsdienstmitarbeiter	X	X	X		
Qualifikationskontrollen	X	X	X	X	X
Ölanlage	X	X	X	X	X
Medienzelle	X	X	X	X	X
Kreditkartenmarkt	X	X	X	X	X
Kreditkartenmarkt		X		X	
Rückzahlungsprioritäten	X	X	X		X

Kreditkartenmarkt	X	X	X	X	
Luxusmarken	X	X	X	X	X
Goldfäden	X	X	X	X	
Luxusautos	X	X	X	X	
Millionärsmesse	X	X	X	X	X
Polizeidirektors	X	X	X	X	X
Busladungen	X	X	X	X	X
Regierungskonvoi	X	X	X	X	X
Bildungsmonitor	X	X	X	X	X
Bildungsexperte	X	X	X	X	X
Risikoschüler	X	X	X		
Nachbarplaneten	X	X	X	X	X
Abendstern	X	X	X	X	X
Parteienstreits	X	X	X	X	X
Folterdebatten	X	X	X		
Popgedächtnis	X	X	X	X	X
Plattenlabel	X	X		X	X
Kriegsreporter	X	X	X	X	X
Kopfgeldjäger	X	X	X	X	X
Wohnnutzung	X	X	X	X	X
Holzkisten	X	X	X	X	X
Museumsdirektor	X	X	X	X	X
Holzkisten	X	X	X	X	X
Auktionsgeschichte	X	X	X	X	X
Museumsdirektor	X	X	X	X	X
Holzkisten	X	X	X	X	X
Acrylfarbe	X	X	X	X	X
Großaktionär	X			X	X
all:	**47**	**47**	**46**	**44**	**43**

Table D.1.: Exact reference matches of compounds, produced by the compound processing systems, which have not occured in the parallel training data.

Bibliography

Adda-Decker, M., Adda, G., and Lamel, L. (2000). Investigating Text Normalization and Pronunciation Variants for German Broadcast Transcription. In *INTERSPEECH'00: Proceedings of the Annual Conference of International Speech Communication Association*, pages 266–269.

Airio, E. (2006). Word Normalization and Decompounding in Mono- and Bilingual IR. *Information Retrieval*, 9(3):249–271.

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F.-J., Purdy, D., Smith, N. A., and Yarowsky, D. (1999). Statistical Machine Translation. Technical report, Final report, JHU summer workshop.

Alfonseca, E., Bilac, S., and Pharies, S. (2008a). Decompounding Query Keywords from Compounding Languages. In *ACL'08: Proceedings of the 46th annual meeting of the Association for Compuational Linguistics, Short Papers (Companion Volume)*, pages 253–256.

Alfonseca, E., Bilac, S., and Pharies, S. (2008b). German Decompounding in a Difficult Corpus. In *CICLING'08: Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 128–139. Springer Verlag.

Avramidis, E. and Koehn, P. (2008). Enriching Morphologically Poor Languages for Statistical Machine Translation. In *ACL'08: Proceedings of the 46th annual meeting of the Association for Compuational Linguistics, Short Papers (Companion Volume)*, pages 763–770.

Badr, I., Zbib, R., and Glass, J. (2008). Segmentation for English-to-Arabic Statistical Machine Translation. In *ACL'08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 153–156. Association for Computational Linguistics.

Bai, M.-H., Chen, K.-J., and Chang, J. S. (2008). Improving Word Alignment by Adjusting Chinese Word Segmentation. In *IJCNLP'08: Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 249–256.

Baldwin, T. and Tanaka, T. (2004). Translation by Machine of Complex Nominals: Getting it Right. In *ACL'04: Proceedings of the Workshop on Multiword Expressions: Integrating Processing of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 24–31. Association for Computational Linguistics.

Baroni, M., Matiasek, J., and Trost, H. (2002). Predicting the Components of German Nominal Compounds. In *Proceedings of the XXth European Conference on Artificial Intelligence*, pages 470–474.

Bergenholtz, H. and Mugdan, J. (1979). *Einführung in die Morphologie*. W. Kohlhammer GmbH.

Berton, A., Fetter, P., and Regel-Brietzmann, P. (1996). Compound Words in Large-vocabulary German Speech Recognition Systems. In *Proceedings of the 4th International Language on Spoken Language Processing*, pages 1165–1168.

Bisazza, A. and Federico, M. (2009). Morphological Pre-processing for Turkish to English Statistical Machine Translation. In *IWSLT'09: Proceedings of the International Workshop on Spoken Language Translation*, pages 129–135.

Botha, J. A., Dyer, C., and Blunsom, P. (2012). Bayesian Language Modelling of German Compounds. In *COLING'12: Proceedings of the 24th International Conference on Computational Linguistics*, pages 341–356.

Braschler, M. and Ripplinger, B. (2004). How Effective is Stemming and Decompounding for German Text Retrieval? *Information Retrieval*, 7(3-4):291–316.

Brown, P. F., Della Pietra, S. A., Della Pietra, V., and Mercer, R. L. (1992). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Brown, R. D. (2002). Corpus-driven Splitting of Compounds. In *TMI'02: Proceedings of the 9th international conference on theoretical and methodological issues in machine translation.*

Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyser. Linguistic Data Consortium (LDC2002L49).

Cap, F., Fraser, A., Weller, M., and Cahill, A. (2014a). How to Produce Unseen Teddy Bears: Improved Morphological Processing of Compounds in SMT. In *EACL'14: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 579–587, Gothenburg, Sweden. Association for Computational Linguistics.

Cap, F., Weller, M., Ramm, A., and Fraser, A. (2014b). CimS – The CIS and IMS joint submission to WMT 2014 translating from English into German. In *ACL'14: Proceedings of the 8th Workshop on Statistical Machine Translation and Metrics MATR of the 52th Annual Meeting of the Association for Computational LinguisticsWMT'14: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 71–78, Baltimore, MD, USA. Association for Computational Linguistics.

Carlberger, J. and Kann, V. (1999). Implementing an Efficient Part-of-Speech Tagger. *Software, Practise and Experience*, 29(9):815–832.

Chahuneau, V., Schlinger, E., Smith, N. A., and Dyer, C. (2013). Translating into Morphologically Rich Languages with Synthetic Phrases. In *EMNLP'13: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1677–1687. ACL.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking. In *ACL'05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, Michigan.

Cherry, C. and Foster, G. (2012). Batch Tuning Strategies for Statistical Machine Translation. In *HLT-NAACL'12: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, volume 12, pages 34–35.

Chung, T. and Gildea, D. (2009). Unsupervised Tokenization for Machine Translation. In *EMNLP'09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 2, pages 718–726. Association for Computational Linguistics.

Clifton, A. and Sarkar, A. (2011). Combining Morpheme-based Machine Translation with Post-processing Morpheme Prediction. In *ACL'11: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 32–42,.

Corston-Oliver, S. and Gamon, M. (2004). Normalizing German and English Inflectional Morphology to Improve Statistical Word Alignment. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*, pages 48–57. Springer Verlag.

Creutz, M. and Lagus, K. (2005). Inducing the Morphological Lexicon of a Natural Language from Unannotated Text. In *AKRR'05: Proceedings of the international and interdisciplinary conference on adaptive knowledge representation and reasoning.*

deGispert, A. and Mariño, J. B. (2008). On the Impact of Morphology in English to Spanish Statistical MT. *Speech Communication*, 50:1034–1046.

Dejean, H., Gaussier, E., Goutte, C., and Yamada, K. (2003). Reducing Parameter Space for Word Alignment. In *HLT-NAACL'03: Proceedings of the Workshop on Building and using parallel texts: data driven machine translation and beyond-Volume 3*, pages 23–26. Association for Computational Linguistics.

Demberg, V. (2006). Letter-to-Phoneme Conversion for a German Text-to-Speech System. Master's thesis, University of Stuttgart, Stuttgart, Germany.

Demberg, V. (2007). A Language-independent Unsupervised Model for Morphological Segmentation. In *ACL'07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 920–927.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

DeNeefe, S., Hermjakob, U., and Knight, K. (2008). Overcoming Vocabulary Sparsity in MT using Lattices. In *AMTA'08: Proceedings of the 8th Biennial Conference of the Association for Machine Translation in the Americas*.

Durgar El-Kahlout, I. and Oflazer, K. (2006). Initial Explorations in English to Turkish Statistical Machine Translation. In *HLT-NAACL'06: Proceedings of the 1st workshop on statistical machine translation of the human language technology conference of the north American chapter of the Association for Computational Linguistics*, pages 7–14. Association for Computational Linguistics.

Durgar El-Kahlout, I. and Yvon, F. (2010). The Pay-offs of Preprocessing for German-English Statistical Machine Translation. In *IWSLT'10: Proceedings of the seventh International Workshop on Spoken Language Translation*, pages 251–258.

Dyer, C. (2009). Using a Maximum Entropy Model to Build Segmentation Lattices for MT. In *HLT-NAACL'09: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 406–414.

Dyer, C. (2010). *A Formal Model of Ambiguity and its Applications in Machine Translation.* PhD thesis, University of Maryland, MD, USA.

Dyer, C., Muresan, S., and Resnik, P. (2008). Generalizing Word Lattice Translation. In *ACL'08: Proceedings of the 46th annual meeting of the Association for Compuational Linguistics*, pages 1012–1020, Columbus, Ohio. Association for Computational Linguistics.

El Isbihani, A., Khadivi, S., Bender, O., and Ney, H. (2006). Morpho-syntactic Arabic Preprocessing for Arabic-to-english Statistical Machine Translation. In *HLT-NAACL'06: Proceedings of the 1st workshop on statistical machine translation of the human language technology conference of the north American chapter of the Association for Computational Linguistics*, pages 15–22. Association for Computational Linguistics.

El Kholy, A. and Habash, N. (2010). Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *LREC'10: Proceedings of the seventh International Conference on Language Resources and Evaluation*.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* MIT Press.

Fishel, M. and Kirik, H. (2010). Linguistically Motivated Unsupervised Segmentation for Machine Translation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Fleischer, W. and Barz, I. (1995). *Wortbildung der deutschen Gegenwartssprache.* Niemeyer, Tübingen.

Fraser, A. (2009). Experiments in Morphosyntactic Processing for Translation to and from German. In *EACL'09: Proceedings of the 4th Workshop on Statistical Machine Translation of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 115–119.

Fraser, A., Weller, M., Cahill, A., and Cap, F. (2012). Modeling Inflection and Word Formation in SMT. In *EACL'12: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674.

Fritzinger, F. and Fraser, A. (2010). How to avoid Burning Ducks: Combining Linguistic Analysis and Corpus Statistics for German Compound Processing. In *ACL'10: Proceedings of the 5th Workshop on Statistical Machine Translation and Metrics MATR of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 224–234.

Fuhrhop, N. (1996). Fugenelemente. In Lang, E. and Zifonun, G., editors, *Deutsch-typologisch*, IDS Jahrbuch, pages 525–550. deGruyter.

Fuhrhop, N. (1998). *Grenzfälle morphologischer Einheiten.* Stauffenburg, Tübingen.

Gao, Q. and Vogel, S. (2008). Parallel Implementations of Word Alignment Tool. In *ACL'08: Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 49–57. Association for Computational Linguistics.

Garera, N. and Yarowsky, D. (2008). Translating Compounds by Learning Component Gloss Translation Models via Multiple Languages. In *IJCNLP'08: Proceedings of the 3rd International Conference on Natural Language Processing*, pages 403–410.

Goldwater, S. and McClosky, D. (2005). Improving Statistical MT through Morphological Analysis. In *HLT-EMNLP '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 676–683, Morristown, NJ, USA. Association for Computational Linguistics.

Habash, N., Gabbard, R., Rambow, O., Kulick, S., and Marcus, M. P. (2007). Determining Case in Arabic: Learning Complex Linguistic Behavior Requires Complex Linguistic Features. In *EMNLP'07: Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing*, pages 1084–1092.

Habash, N. and Metsky, H. (2008). Automatic Learning of Morphological Variations for Handling Out-of-vocabulary Terms in Urdu-English Machine Translation. In *AMTA'08: Proceedings of the Association for Machine Translation in the Americas.*

Habash, N. and Rambow, O. (2005). Arabic Tokenization, Part-of-Speech-Tagging and Morphological Disambiguation in One Fell Swoop. In *ACL'05: Proceedings of the 43rd annual meeting of the Association for Compuational Linguistics*, pages 573–580. Association for Computational Linguistics.

Habash, N. and Sadat, F. (2006). Arabic Preprocessing Schemes for Statistical Machine Translation. In *NAACL-HLT'06: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 49–52.

Hardmeier, C., Bisazza, A., and Federico, M. (2010). FBK at WMT 2010: Word Lattices for Morphological Reduction and Chunk-based Reordering. In *ACL'10: Proceedings of the 5th Workshop on Statistical Machine Translation and Metrics MATR of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 88–92. Association for Computational Linguistics.

Hausser, R. (1996). *Linguistische Verifikation: Dokumentation zur Ersten Morpholympics 1994*. Niemeyer.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *EMNLP'11: Proceedings of the 6th workshop on statistical machine translation within the 8th Conference on Empirical Methods in Natural Language Processing*, pages 187–197.

Heid, U., Säuberlich, B., and Fitschen, A. (2002). Using Descriptive Generalisations in the Acquisition of Lexical Data for Word Formation. In *LREC*.

Holz, F. and Biemann, C. (2008). Unsupervised and Knowledge-free Learning of Compound Splits and Periphrases. In *Computational Linguistics and Intelligent Text Processing*, volume 4919/2008, pages 117–127. Spinger Verlag, Berlin/Heidelberg.

Hutchins, W. J. and Somers, H. L. (1992). *An Introduction to Machine Translation*, volume 362. Academic Press London.

Karttunen, L. (2001). Applications of Finite-state Transducers in Natural Language Processing. In *Implementation and application of automata*, pages 34–46. Springer.

Karttunen, L. and Beesley, K. R. (2001). A short History of Two-level Morphology. *ESSLLI-2001 Special Event titled" Twenty Years of Finite-State Morphology*.

Keshava, S. and Pitler, E. (2006). A Simpler, Intuitive Approach to Morpheme Induction. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 31–35.

Koehn, P. (2005). Europarl: a Parallel Corpus for Statistical Machine Translation. In *MT Summit'05: Proceedings of the 10th machine translation summit*, pages 79–86.

Koehn, P. (2009). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P. and Hoang, H. (2007). Factored Translation Models. In *EMNLP'07: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 868–876.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL'07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Demonstration Session*, pages 177–180.

Koehn, P. and Knight, K. (2003). Empirical Methods for Compound Splitting. In *EACL '03: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–193, Morristown, NJ, USA. Association for Computational Linguistics.

Koskenniemi, K. (1983). *Two-level Morphology: a General Computational Model for Word-form Recognition and Production.* PhD thesis, University of Helsinki, Helsinki, Finland.

Koskenniemi, K. and Haapalainen, M. (1996). GERTWOL – Lingsoft Oy. In Hauser, R., editor, *Linguistische Verifikation, Dokumentation zur ersten Morpholympics 1994*, pages 121–140. Niemeyer, Tübingen.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML'01: Proceedings of the 18th International Conference on Machine Learning.*

Langer, S. (1998). Zur Morphologie und Semantik von Nominalkomposita. In *KONVENS'98: Tagungsband der 4. Konferenz zur Verarbeitung natürlicher Sprache*, pages 83–97.

Larson, M., Willett, D., Köhler, J., and Rigoll, G. (2000). Compound Splitting and Lexical Unit Recombination for Improved Performance of a Speech Recognition System for German Parliamentary Speeches. In *ICSLP'00: Proceedings of the 6th International Conference on Spoken Language Processing*, pages 945–948.

Lavie, A. and Agarwal, A. (2007). Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgements. In *ACL'07: Proceedings of the 2nd Workshop on Statistical Machine Translation within the 45th Annual Meeting of the Association for Computational Linguistics*, pages 228–231.

Lee, Y.-S. (2004). Morpholocial Analysis for Statistical Machine Translation. In *HLT-NAACL'04: Proceedings of the human language technology conference of the north American chapter of the Association for Computational Linguistics, Short Papers*, pages 57–60. Association for Computational Linguistics.

Lee, Y.-S., Papineni, K., Roukos, S., Emam, O., and Hassan, H. (2003). Language Model based Arabic Word Segmentation. In *ACL'03: Proceedings of the 41st Annual Meeting of the Association for Compuational Linguistics*, pages 399–406. Association for Computational Linguistics.

Lopez, A. (2008). Statistical Machine Translation. *ACM Computing Surveys*, 40(3):1–49.

Luong, M.-T., Nakov, P., and Kan, M.-Y. (2010). A Hybrid Morpheme-Word Representation for Machine Translation of Morphologically Rich Languages. In *EMNLP'10: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 148–157.

Ma, Y., Stroppa, N., and Way, A. (2007). Bootstrapping Word Alignment via Word Packaging. In *ACL'07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 304–311.

Macherey, K., Dai, A. M., Talbot, D., Popat, A. C., and Och, F. (2011). Language-independent Compound Splitting with Morphological Operations. In *ACL '11: Proceedings of the 49th annual meeting of the Association for Computational Linguistics*, pages 1395–1404.

Marek, T. (2006). Analysis of German Compounds using Weighted Finite State Transducers. Bachelor's thesis, Eberhard-Karls-Universität Tübingen, Germany.

McDonald, R. and Pereira, F. (2005). Identifying Gene and Protein Mentions in Text using Conditional Random Fields. *BMC bioinformatics*, 6(Suppl 1):S6.

Minkov, E., Toutanova, K., and Suzuki, H. (2007). Generating Complex Morphology for Machine Translation. In *ACL '07: Proceedings of the 45th annual meeting of the Association for Computational Linguistics*, pages 128–135.

Monz, C. and de Rijke, M. (2001). Shallow Morphological Analysis in Monolingual Information Retrieval for Dutch, German and Italian. In *CLEF'01: Proceedings of the 2nd workshop of the cross-language evaluation forum*, pages 262–277, London, UK. Springer-Verlag.

Moulinier, I., McCulloh, J. A., and Lund, E. (2001). West Group at CLEF 2000: Non-English Monolingual Retrieval. In Peters, C., editor, *Cross-language Information Retrieval and Evaluation*, volume 2069 of *Lecture Notes in Computer Science*, pages 253–260. Springer.

Naradowsky, J. and Toutanova, K. (2011). Unsupervised Bilingual Morpheme Segmentation and Alignment with Context-rich Hidden Semi-Markov Models. In *ACL'11: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 895–904. Association for Computational Linguistics.

Nießen, S. and Ney, H. (2000). Improving SMT Quality with Morpho-syntactic Analysis. In *COLING'00: Proceedings of the 18th International Conference on Computational Linguistics*, pages 1081–1085. Morgan Kaufmann.

Nießen, S. and Ney, H. (2004). Statistical Machine Translation with Scarce Resources using Morpho-syntactic Information. *Computational Linguistics*, 30(2):181–204.

Novák, A. (2009). MorphoLogic's Submission for the WMT 2009 Shared Task. In *EACL'09: Proceedings of the 4th Workshop on Statistical Machine Translation of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 155–159. Association for Computational Linguistics.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *ACL'03: Proceedings of the 41st Annual Meeting of the Association for Compuational Linguistics*, pages 160–167.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Oflazer, K. (1994). Two-level Description of Turkish Morphology. *Literary and linguistic computing*, 9(2):137–148.

Oflazer, K. and Durgar El-Kahlout, I. (2007). Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. In *ACL'07: Proceedings of the 2nd workshop on statistical machine translation of the 45th annual meeting of the Association for Computational Linguistics*, pages 25–32.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A Method for Automatic Evaluation of Machine Translation. In *ACL'02: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Popović, M. and Ney, H. (2004). Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. In *LREC'04: Proceedings of the 4th international conference on language ressources and evaluation*, pages 1585–1588.

Popović, M., Stein, D., and Ney, H. (2006). Statistical Machine Translation of German Compound Words. In *FinTAL'06: Proceedings of the 5th International Conference on Natural Language Processing*, pages 616–624. Springer Verlag.

Quernheim, D. and Cap, F. (2014). Large-scale Exact Decoding: The IMS-TTT Submission to WMT14. In *ACL'14: Proceedings of the 8th Workshop on Statistical Machine Translation and Metrics MATR of the 52th Annual Meeting of the Association for Computational LinguisticsWMT'14: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 163–170.

Rackow, U., Dagan, I., and Schwall, U. (1992). Automatic Translation of Noun Compounds. In *COLING'92: Proceedings of the 14th International Conference on Computational Linguistics*, pages 1249–1253.

Schiller, A. (2005). German Compound Analysis with wfsc. In *FSMNLP'05: Proceedings of the Fifth Internation Workshop of Finite State Methods in Natural Language Processing.*

Schmid, H. (1994). Probabilistic Part-of-speech Tagging using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Schmid, H. (2004). Efficient Parsing of Highly Ambiguous Context-free Grammars with Bit Vectors. In *COLING'04: Proceedings of the 20th international conference on Computational Linguistics*, page 162. Association for Computational Linguistics.

Schmid, H., Fitschen, A., and Heid, U. (2004). SMOR: A German Computational Morphology Covering Derivation, Composition and Inflection. In *LREC '04: Proceedings of the 4th Conference on Language Resources and Evaluation*, pages 1263–1266.

Sereewattana, S. (2003). Unsupervised Segmentation for Statistical Machine Translation. Master's thesis, University of Ediburgh, Edinburgh, UK.

Shannon, C. E. (2001). A Mathematical Theory of Communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

Spies, M. (1995). A Language Model for Compound Words in Speech Recognition. In *EUROSPEECH '95: Proceedings of the 4th European Conference on Speech Communication and Technology*, pages 1767–1770.

Stolcke, A. (2002). SRILM – An Extensible Language Modelling Toolkit. In *ICSLN'02: Proceedings of the international conference on spoken language processing*, pages 901–904.

Stymne, S. (2008). German Compounds in Factored Statistical Machine Translation. In *GoTAL '08: Proceedings of the 6th International Conference on Natural Language Processing*, pages 464–475. Springer Verlag.

Stymne, S. (2009). A Comparison of Merging Strategies for Translation of German Compounds. In *EACL '09: Proceedings of the Student Research Workshop of the 12th conference of the European Chapter of the Association for Computational Linguistics*, pages 61–69.

Stymne, S. and Cancedda, N. (2011). Productive Generation of Compound Words in Statistical Machine Translation. In *EMNLP'11: Proceedings of the 6th Workshop on Statistical Machine Translation and Metrics MATR of the conference on Empirical Methods in Natural Language Processing*, pages 250–260. Association for Computational Linguistics.

Stymne, S. and Holmqvist, M. (2008). Processing of Swedish Compounds for Phrase-based Statistical Machine Translation. In *EAMT '08: Proceedings of the 12th annual conference of the European Association for machine translation*, pages 180–189.

Talbot, D. and Osborne, M. (2006). Modelling Lexical Redundancy for Machine Translation. In *COLING'06/ACL'06: Proceedings of the 21th International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 969–976. Association for Computational Linguistics.

Tanaka, T. and Baldwin, T. (2003). Noun-noun Compound Machine Translation: A Feasibility Study on Shallow Processing. In *ACL'03: Proceedings of the workshop on Multiword expressions: analysis, acquisition and treatment of the 41st Annual Meeting of the Association for Compuational Linguistics*, pages 17–24. Association for Computational Linguistics.

Toutanova, K. and Suzuki, H. (2007). Generating Case Markers in Machine Translation. In *HLT-NAACL'07: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 49–56. Association for Computational Linguistics.

Toutanova, K., Suzuki, H., and Ruopp, A. (2008). Applying Morphology Generation Models to Machine Translation. In *ACL'08: Proceedings of the 46th annual meeting of the Association for Compuational Linguistics*, pages 514–522. Association for Computational Linguistics.

Ueffing, N. and Ney, H. (2003). Using POS Information for Statistical Machine Translation into Morphologically Rich Languages. In *EACL'03: Proceedings of the 10th conference on European chapter of the Association for Computational Linguistics*, pages 347–354. Association for Computational Linguistics.

Vauquois, B. (1968). A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Mechanical Translation. In *IFIP Congress (2)'68*, pages 1114–1122.

Čmejrek, M., Cuřín, J., and Havelka, J. (2003). Czech-English Dependency-parsed Machine Translation. In *EACL '03: Proceedings of the 10th conference of the European chapter of the Association for Computational Linguistics*, pages 83–90.

Virpioja, S., Väyrynen, J. J., Creutz, M., and Sadeniemi, M. (2007). Morphology-aware Statistical Machine Translation based on Morphs Induced in an Unsupervised Manner. In *MT Summit '07: Proceedings of the 11th Machine Translation Summit*, pages 491–498,[**CS**,**LM**].

Weller, M. (2009). Separate Morphologiebehandlung als Methode zur Verbesserung statistischer maschineller Übersetzung. Master's thesis, Universität Stuttgart, Stuttgart, Germany.

Weller, M., Cap, F., Müller, S., Schulte im Walde, S., and Fraser, A. (2014). Distinguishing Degrees of Compositionality in Compound Splitting for Statistical Machine Translation. In *ComAComA'14: Proceedings of the First Workshop on Computational Approaches to Compound Analysis at COLING 2014*.

Weller, M. and Heid, U. (2012). Analyzing and Aligning German Compound Nouns. In *LREC'12: Proceedings of the 8th international conference on language ressources and evaluation*. European Language Resources Association (ELRA).

Xu, J., Matusov, E., Zens, R., and Ney, H. (2005). Integrated Chinese Word Segmentation in Statistical Machine Translation. In *IWSLT'05: Proceedings of the International Workshop on Spoken Language TranslationIWSLT*, pages 131–137.

Yang, M. and Kirchhoff, K. (2006). Phrase-based Backoff Models for Machine Translation of Highly Inflected Languages. In *EACL'06: Proceedings of the 11th conference of the European chapter of the Association for Computational Linguistics*, pages 41–48.

Yun, B.-H., Lee, H., and Rim, H.-C. (1995). Analysis of Korean Compound Nouns using Statistical Information. In *ICCPOL'95: Proceedings of the International Conference on Computer Processing of Oriental Languages*, pages 76–79.

Yuret, D. and Türe, F. (2006). Learning Morphological Disambiguation Rules for Turkish. In *HLT-NAACL'06: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 328–334. Association for Computational Linguistics.

Zhang, J., Gao, J., and Zhou, M. (2000). Extraction of Chinese Compound Words - An Experimental Study on a Very Large Corpus. In *Second Chinese Language Processing Workshop*, pages 132–139, Hong Kong, China. Association for Computational Linguistics.

Zollmann, A., Venugopal, A., and Vogel, S. (2006). Bridging the Inflection Morphology Gap for Arabic Statistical Machine Translation. In *HLT-NAACL'06: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 201–204. Association for Computational Linguistics.