Typologically Robust Statistical Machine Translation

Typologically Robust Statistical Machine Translation

Understanding and Exploiting Differences and Similarities Between Languages in Machine Translation

Institute for Logic, Language and Computation

Joachim Daiber

Joachim Daiber

Typologically Robust Statistical Machine Translation

Understanding and Exploiting Differences and Similarities Between Languages in Machine Translation

Joachim Daiber

ILLC Dissertation Series DS-2018-05

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

This work was supported by the EXPERT Initial Training Network of the European Union's 7th Framework Programme (EXPloiting Empirical appRoaches to Translation, grant agreement No. 317471) and Quality Translation 21 (QT21), European Union's Horizon 2020 research and innovation programme (grant agreement No. 645452).

Cover photo taken by the author in Bregenz, Austria on October 15, 2016.
Printed and bound by Ipskamp Printing.

ISBN: 978-94-028-0947-3

Typologically Robust Statistical Machine Translation

Understanding and Exploiting Differences and Similarities Between Languages in Machine Translation

Academisch Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 20 maart 2018, te 12.00 uur

door

Joachim Daiber

geboren te Bad Waldsee, Duitsland

Promotiecommissie:

Promotor:	Prof. dr. K. Sima'an	Universiteit van Amsterdam
Co-promotor:	Dr. W. Ferreira Aziz	Universiteit van Amsterdam
Overige leden:	Prof. dr. J. van Genabith	Universität des Saarlandes
	Prof. dr. G.J.M. van Noord	Rijksuniversiteit Groningen
	Dr. C. Monz	Universiteit van Amsterdam
	Dr. W.H. Zuidema	Universiteit van Amsterdam
	Prof. dr. H.J. Honing	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

To my parents and my brother.

Contents

Acknowledgments

I am deeply indebted to my supervisor Khalil Sima'an for his guidance, mentorship and patience over these past 4 years. By offering me this PhD position, he gave me the opportunity to pursue my interests in natural language processing while working on exciting topics with a great team of fellow researchers. Our weekly meetings provided fertile ground for new ideas and Khalil helped me connect the dots where at first I only saw unrelated problems. I have always looked forward to our meetings and have always left them with new insights. I am very thankful to Khalil for giving me the freedom to develop my interests beyond machine translation and for providing his students with an environment in which we were shielded from many issues not directly related to our research. I was also lucky to have a great co-supervisor in Wilker Aziz, to whom I could always turn with technical and formal questions and whose feedback and support helped me stay motivated during the final phases of my PhD.

The ILLC is a great environment for PhD students and I feel privileged to have been able to enjoy these years in Amsterdam. This PhD journey would have been much more difficult without the continued support from so many people at the ILLC, including Jenny Batson, Sonja Smets, Raquel Fernández, Karine Gigengack, Tanja Kassenaar, Fenneke Kortenbach and Debbie Klaassen. I fondly look back at many stimulating conversations I had over lunch and at ILLC events with Jelle Zuidema, Benno van den Berg, Ivan Titov, Miguel Rios Gaona, Diego Marcheggiani, and many others. Helping Khalil and Ivan teach their natural language processing classes was a great learning experience and made me a better researcher and teacher.

My personal journey into natural language processing began in 2009 with an internship at a startup in Berlin. I want to thank Martin Hirsch, who gave me the chance to gain practical experience in the field even though I still lacked much of the necessary background, and Georg Rehm, from whom I learnt a great deal during this time and later at DFKI in Berlin. While in Berlin, I also first met Pablo Mendes, with whom I collaborated on information extraction. My interests during my Master's and PhD lead me to parsing and translation but my fascination with information extraction has always remained and I am excited to be able to work on it again today. Pablo continued to push

me in the right direction and I am glad that he convinced me to intern at Lattice Data, where I met many inspiring people (Chris Ré, Feng Niu, Xiao Ling, Michele Banko, Art Clarke, ...) and which turned out to be a great adventure.

My PhD would also not have been the same without all the friends and colleagues I shared it with: Samira Abnar, Laura Aina, Sophie Arnoult, Joost Bastings, Sirin Botan, Andreas van Cranenburgh, Desmond Elliott, Stella Frank, LiFeng Han, Cuong Hoang (whose smartness I would like to highlight; and not only because he won a bet on the outcome of the US election), Dieuwke Hupkes, Bushra Jawaid, Amir Kamran, Ehsan Khoddammohammadi, Gideon Maillette de Buy Wenniger, Diego Marcheggiani, Gert-Jan Munneke, Răzvan Pavel, Philip Schulz, Josefine Ulbrich, Sara Veldhoen, Anya Zaretskaya, ...

Six years ago, when I had just started my Master's in Prague, I first met Miloš Stanojević and Ke Tran. I can still remember our first conversations: Ke told me how awesome neural networks are and Miloš reported on the quality of Czech beer. While many things changed during these six years, I am happy that Ke and Miloš (just like beer and neural networks) remained constant parts of my life. After following them to Amsterdam, I soon met Raquel Garrido Alhama and Phong Le. Being a PhD student was not always easy, but Miloš, Ke, Raquel and Phong made my time in Amsterdam into something special. I am grateful for all the basketball games we played, great beers we had, for our gym sessions that were often just a thinly veiled excuse to relax in the sauna, for our endless discussions and for the trips we did together.

Finally, I want to thank my family in Germany for all the love and support they gave me over this long journey: Danke Mama, Papa, Sebastian, Maria und Wilhelm!

San Francisco
January, 2018.

Joachim Daiber

Chapter 1

Introduction

1.1 Introduction

Machine translation is a central task in natural language processing research, which touches on many related tasks and subfields of this area of research. Throughout the history of the field, research in machine translation has often been constrained by the limited availability of suitable training data for building translation models. In recent years such data has increasingly become available but approaches to machine translation still frequently show a bias towards the characteristics of the language pairs for which data has been more readily available. Phrase-based machine translation, which has been at the core of machine translation research for the past years, for example, relies partly on the assumption that the word order of the languages it is applied to is relatively fixed, thus allowing the extraction of meaningful sequences of words without data sparsity. However, as data for a broader set of language pairs has become available, it has become apparent that cross-linguistic differences can have a significant influence on the quality of machine translation.

In this thesis, we investigate to which extent the characteristics of the source and target language influence translation quality and whether such characteristics can be utilized to produce more principled translation models for a broader range of language pairs. In linguistic theory, the characteristics and differences between languages are studied in the field of linguistic typology. Two areas of linguistic typology, word order and morphological complexity, are central to machine translation and the properties of languages in these two areas significantly influence to which extent the basic assumptions of many machine translation systems hold.

1.2 Objective

How do typological differences in languages influence the performance of machine translation systems? And how can typological differences be modeled in order to improve existing machine translation systems? In this thesis, we examine these two questions and argue that to create more principled models for machine translation, we must take into account knowledge about typological differences between languages, especially in the areas of word order and morphology. Integrating knowledge about the range of possible differences in languages is expected to (1) improve translation quality for languages for which the standard models do not perform well and (2) improve how well machine translation models generalize to typologically diverse languages.

While our findings are not limited to this approach, in this thesis we focus mainly on statistical machine translation using phrase-based models. Phrase-based machine translation (Koehn et al., 2003; Och, 2002) has been at the core of state-of-the-art translation systems in recent years. Phrase-based models match sequences of words in the source sentence with observed translations and combine the observed target sequences into sentences by reordering and scoring them using bilingual and monolingual features.

We focus in particular on two areas central to machine translation:

- **Word order:** Determining the most suitable order of words and phrases in the target sentence is a crucial task in machine translation. Preordering, one of the established methods for this task, has found wide adoption but has not benefited all language pairs equally. In preordering, the source sentence is ordered in the predicted target order, which relieves the translation system from having to perform potentially costly long-distance reordering operations and allows a more thorough exploration of the space of word order permutations since it does not have to take into account full translation. How do typological aspects such as word order freedom and morphological complexity influence machine translation in general? And in particular, do such aspects impede the generalization of approaches such as preordering to typologically diverse target languages?
- **Word formation:** Word formation processes vary in productivity from language to language. Since phrase-based machine translation relies on combining words into larger units for translation, productive morphological processes can interfere with the translation process. Can the limits in the ability of statistical translation systems to handle productive word formation processes be overcome by making such differences overt to the translation system? We explore this idea for the two crucial word formation processes of inflectional morphology and compounding.

How can we ensure that the benefits of machine translation models are not overly concentrated on specific types of languages but will apply to a wide range of typologically diverse languages? In this thesis we examine word order and morphological complexity, two areas in which languages show significant divergences, and propose models that are robust to such typological differences.

1.3 Contribution

We begin by examining the area of word order. Unlike English, which exhibits relatively rigid word order, many languages allow for more freedom in the order of words and constituents. How does this word order freedom influence machine translation? As our first contribution, we examine this question by considering one particular method to deal with word order differences, namely preordering. Do the assumptions made in preordering hold for free word order languages? We first discuss the common definitions of word order freedom in the linguistic literature and find that few quantitative measures applicable to machine translation exist. Thus, we introduce an entropy-based measure to quantify word order freedom based on source dependency trees and parallel sentences. We find that the assumptions of preordering models are ill-fitted for free word order and morphologically rich target languages (Chapter 4).

How can the typological robustness of preordering be improved? We propose to pass a space of potential word order choices instead of a single-best word order prediction to the machine translation system. In Chapter 5, we perform a case study with a morphologically rich target language with relatively free word order, specifically using the language pair English–German, and find that this approach provides great potential for a more principled treatment of such language pairs. Given the potential of the idea of passing the preordering model's space of word order choices to the translation system, how can this space be represented to allow for efficient integration into the machine translation system? In Chapter 6, we propose the use of word order permutation lattices to integrate the space of potential word order choices with the translation system, which can then decide on a final word order while taking into account further relevant information, such as lexical choice. We use our entropy measure of word order freedom to select two target languages from opposite ends of the word order freedom spectrum — Japanese as an instance of a strict word order language and German as an instance of a language with less strict word order — and show that word order permutation lattices provide a suitable representation for both target languages. This demonstrates that permutation lattices of potential word order choices provide a typologically robust solution to integrating preordering models into machine translation.

The second area of significant typological differences in machine translation is morphological complexity. Here, we make contributions to two central areas. First, when translating from a morphologically impoverished to a morphologically rich target language, the typological differences of the language pair cause several challenges for phrase-based machine translation systems, including data sparsity and the inability of phrase-based models to reliably enforce morphological agreement over long distances. Is it possible to bridge such typological differences in morphological complexity? We propose to enrich the source language with the morphological attributes required to form the correct target words. In Chapter 7, we show that the morphological attributes which are helpful for finding the correct target word forms can be learnt from parallel data and that predicting such attributes on the source side, in a similar fashion to

preordering, can enable the translation system to perform better lexical choice.

Apart from inflectional morphology, which is the focus of Chapter 7, other productive word formation processes, such as compounding, complicate machine translation. A phrase-based machine translation system which during training has observed Spanish "Estación Oeste" (west station) and the word "Este" (east) can reasonably deduce the translation "Estación Este" for "east station," while a similar system having observed the German word "Westbahnhof" (west station) and the word "Ost" (east) would require knowledge about the internal structure of the observed word "Westbahnhof" to form the correct translation "Ostbahnhof." In order to make the translation system typologically robust and to allow the required generalization for compounding languages such as German, this internal structure would have to be made explicit. In Chapter 8, we introduce a method to surface the internal structure of compound words by splitting them into their meaning-carrying parts. Our method is unsupervised and relies on semantic analogies ("bookshop is to shop as bookshelf is to shelf') based on contextual representations of words obtained from large monolingual corpora (word embeddings).

This thesis focuses on the influence of language characteristics on machine translation. If the similarities and differences between languages can indeed be captured with a small set of parameters, as linguistic typology and various linguistic theories suggest, then models for natural language processing should, beyond just bridging the performance gaps between typologically diverse languages, also be able to benefit from this insight and the knowledge collected to support it. In the area of word order, for instance, this would enable models with better generalization and requiring less training data. Can linguistic typology serve as a source of knowledge to guide reordering models and to facilitate universal reordering models applicable to multiple target languages? In Chapter 9, we examine this question: We show that typological information collected by linguists in the World Atlas of Language Structures (WALS), when combined with neural network techniques, can be used to build universal reordering models. These models perform well on a typologically diverse set of target languages and can choose automatically, which aspects of linguistic typology to pay attention to when predicting the word order for a particular target language.

In summary, we make the following contributions:

- We examine the typological robustness of preordering and show that producing a space of potential word order choices in conjunction with word order permutation lattices provides a suitable solution for both strict and free word order target languages (Daiber and Sima'an, 2015a; Daiber et al., 2016a, Chapters 4–6).
- We show how morphologically impoverished source languages can be enriched with unexpressed morphological attributes in order to bridge typological differences when translating into morphologically rich languages (Daiber and Sima'an, 2015b, Chapter 7).
- We show that distributional semantics in the form of word embeddings can be used to split compounds into their meaning-carrying components, thus allowing

phrase-based translation models to work with comparable compositional translation units in the source and target language (Daiber et al., 2015, Chapter 8).

- We show that linguistic typology can serve as a source of knowledge to guide reordering models and to facilitate universal reordering models applicable to multiple target languages (Daiber et al., 2016b, Chapter 9).

1.4 Overview

The rest of this thesis is organized as follows:

Part I: Background

- **Chapter 2:** The chapter contributes a comprehensive overview of the objective of machine translation research and the three major categories of technical approaches to the problem. We provide a summary of both classical approaches and neural machine translation and introduce the preliminaries of statistical machine translation, including word alignment, phrase-based models, decoding and reordering. While discussing these approaches, we highlight the structure that each approach imposes on the translation process.
- **Chapter 3:** In the second chapter, we discuss more specifically how linguistic structure is represented in machine translation. We then introduce the area of linguistic typology and discuss how it relates to machine translation and natural language processing. We examine two areas of linguistic typology, word order and morphology, in more detail as these two areas will form the areas of focus for Part II and III of this thesis.

Part II: Word Order Freedom

- **Chapter 4**: We study the relationship between typological aspects of a language pair, such as the word order freedom of the target language, and the effectiveness of preordering in statistical machine translation. We first provide an overview of current approaches to preordering and examine the linguistic motivations and limitations of the technique. We find that the assumptions of preordering can be insufficient for morphologically rich and free word order languages. While individual word order differences and morphological complexity are well-studied topics in linguistic theory, the notion of word order freedom is rarely addressed in a quantifiable way. To measure the word order freedom of languages in a quantitative manner, we therefore introduce a novel entropy measure which assesses how difficult it is to determine word order given a source sentence and its syntactic analysis. This measure, which we call bilingual head direction entropy, will enable us to examine the influence of word order freedom on the effectiveness of preordering in more detail in the following chapters.

The content of this chapter is based on the following publications:
Joachim Daiber and Khalil Sima'an. *Delimiting Morphosyntactic Search Space with Source-Side Reordering Models*. In 1st Deep Machine Translation Workshop, 2015.
Joachim Daiber, Miloš Stanojević, Wilker Aziz, and Khalil Sima'an. *Examining the Relationship Between Preordering and Word Order Freedom in Machine Translation*. In First Conference on Machine Translation, 2016.

- **Chapter 5:** We examine the question whether for morphologically rich and free word order target languages, models without any notion of morphology can be used as a means to delimit the search space for a machine translation system to a set of potential word order predictions instead of committing to just a single best order. We propose a novel preordering model based on a popular preordering algorithm (Lerner and Petrov, 2013), which is able to produce both n-best word order predictions as well as distributions over possible word order choices in the form of a lattice. We further show that the integration of non-local language model features can be beneficial for the model's preordering quality and evaluate the space of potential word order choices the model produces.
  The content of this chapter is based on the following publication:
  Joachim Daiber and Khalil Sima'an. *Delimiting Morphosyntactic Search Space with Source-Side Reordering Models*. In 1st Deep Machine Translation Workshop, 2015.

- **Chapter 6:** We address preordering for two target languages at the far ends of the word order freedom spectrum, German and Japanese. For languages with more word order freedom, attempting to predict a single word order given only the source sentence seems less suitable; therefore, we examine solutions which fit both strict word order and free word order target languages. In Chapter 5, we observed that delimiting the space of word order choices provides a potential solution for free word order target languages and that non-local features can support the preordering model in making good word order choices. A more general approach to this initial exploration is to pass the uncertainty of the preordering model on to the machine translation decoder, which can then perform decisions while taking into account a broader set of signals. Thus, we examine lattices of n-best word order predictions as a unified representation for typologically diverse target languages. We present an effective solution to the resulting technical issue of how to select a suitable source word order from the lattice during training. Our experiments show that lattices are crucial for good empirical performance for languages with freer word order (English–German) and can provide additional improvements for fixed word order languages (English–Japanese).
  The content of this chapter is based on the following publication:
  Joachim Daiber, Miloš Stanojević, Wilker Aziz, and Khalil Sima'an. *Examining the Relationship Between Preordering and Word Order Freedom in Machine Translation*. In First Conference on Machine Translation, 2016.

Part III: Morphological Complexity

- **Chapter 7:** When translating from a morphologically impoverished to a morphologically rich language, the typological differences of the language pair cause challenges for phrase-based machine translation systems. In this chapter, we examine whether such typological differences can be reduced by enriching the source language with the missing morphological attributes. We present a translation pipeline consisting of two steps: first, the source string is enriched with target morphological features and then fed into a translation model which performs reordering and chooses lexical items matching the provided morphological features. After performing experiments to test the merit of this proposal, we present a model for predicting target morphological features on the source string and its predicate-argument structure and address two major technical challenges: (1) How can we determine which morphological features should be predicted for a specific language pair? and (2) How can predicted morphological features be integrated into the phrase-based model so that it can also be trained on morphological features from the parallel data for a more efficient pipeline? Finally, we evaluate the approach on an English–German translation task and find promising improvement over the baseline phrase-based system.
  The content of this chapter is based on the following publication:
  Joachim Daiber and Khalil Sima'an. *Machine Translation with Source-Predicted Target Morphology*. In 15th Machine Translation Summit, 2015.
- **Chapter 8:** Compounding is a highly productive word formation process in some languages that is often problematic for natural language processing applications. In this chapter, we investigate whether distributional semantics in the form of word embeddings can enable more semantically motivated processing of compounds than standard string-based methods. We present an unsupervised approach that exploits regularities in the semantic vector space (based on analogies such as "bookshop is to shop as bookshelf is to shelf") to produce compound analyses of high quality. A subsequent compound splitting algorithm based on these analyses is highly effective, particularly for ambiguous compounds. German–English machine translation experiments show that this semantic analogy-based compound splitter leads to better translations than a commonly used frequency-based method.
  The content of this chapter is based on the following publication:
  Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. *Splitting Compounds by Semantic Analogy*. In 1st Deep Machine Translation Workshop, 2015.

Part IV: Linguistic Typology as a Knowledge Source

- **Chapter 9:** In this chapter, we examine how linguistic typology itself can be used as a rich source of information in machine translation. In particular, we explore the idea of building a universal reordering model from English to a large number of target languages. To build this model, we exploit typological features of word order for a large number of target languages together with source (English) syntactic features. We train a single model on a combined parallel corpus representing all (22) involved language pairs. Apart from empirically demonstrating the value provided by typological descriptions of language, our proposed method can produce word order predictions for a broad range of languages, including language pairs with little or no parallel data. When the universal reordering model is used for preordering followed by monotone translation (no reordering inside the decoder), our experiments show that this pipeline gives comparable or improved translation performance with a phrase-based baseline for a large number of language pairs (12 out of 22) from diverse language families.
  The content of this chapter is based on the following publication:
  Joachim Daiber, Miloš Stanojević, and Khalil Sima'an. *Universal Reordering via Linguistic Typology*. In COLING 2016.
- **Chapter 10:** We present a summary of the thesis and our conclusions.

Part I

Background

Chapter 2

Approaches to Machine Translation

The following two chapters will provide a brief overview of the background on various areas of linguistics, computer science and, in particular, machine translation research that this thesis will touch on. We will begin with an overview of recent research into machine translation and will pay particular attention to how syntactic and morphological aspects of language are treated in these approaches. In the second part of this overview, we will discuss in detail how linguistic structure is handled in approaches to machine translation and will provide an introduction to research in linguistic typology and its use in natural language processing.

2.1 Overview

In 1954 Leon Dostert, one of the researchers involved in the earliest attempts at machine translation, confidently exclaimed that "five, perhaps three years hence, interlingual meaning conversion by electronic process in important functional areas of several languages may well be an accomplished fact."[1] Reality did not follow this ambitious plan. Sixty years later, the fields of machine translation and natural language processing have seen significant progress. However, progress often came from unexpected directions and the problem of machine translation is still far from solved today.

Significant amounts of work have been performed in various basic approaches to machine translation including example-based machine translation, rule-based machine translation and statistical machine translation. We will focus here mainly on statistical approaches to machine translation. This section will summarize various types of basic

[1] A copy of the full press release by the International Business Machines Corporation (IBM) dated January 8, 1954 can be found at `https://www-03.ibm.com/ibm/history/exhibits/701/701_translator.html`.

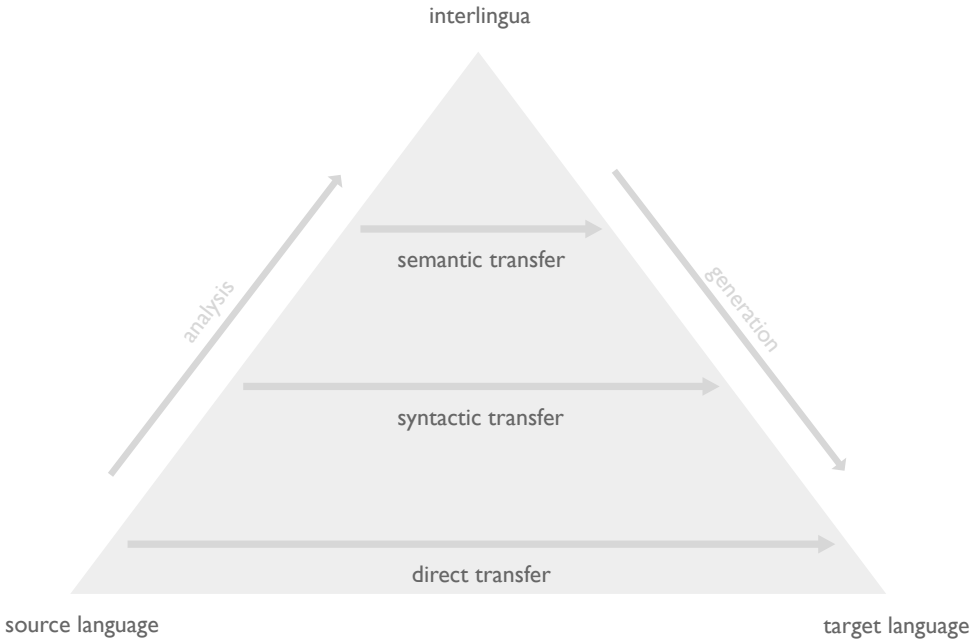


Figure 2.1: Vauquois' triangle.

structures that modern approaches to machine translation assume and, subsequently, we will highlight how these approaches handle difficulties posed by natural language syntax and morphology.

Approaches to machine translation vary widely in their basic assumptions and in the techniques they utilize for training and translation. Nevertheless, it is helpful to categorize and compare these approaches based on their level of structural abstraction, their assumptions about natural language and the level of linguistic analysis at which they place transfer from source to target language.

2.2 Classical Approaches to Machine Translation

The Vauquois triangle (Vauquois, 1968) was originally introduced to discuss classical (i.e., rule-based) approaches to machine translation, but it remains a helpful abstraction for thinking about structure in machine translation to this day. Figure 2.1 illustrates the idea: the source language on the bottom left is transformed into the target language on the bottom right. This transformation can take place at various levels of abstraction.

The form of the pyramid illustrates an important property of these systems: if you consider the length of the path traveled through the pyramid from the source language to the target language as a measure of cost or difficulty, the pyramid illustrates that the cost of transforming the source language into the target language on the level of direct transfer is greater than for all other approaches, while it does not incur costs for analysis

and generation. Conversely, when moving up the pyramid, the cost of transfer diminishes while the cost of analysis (in the source language) and the cost of generation (in the target language) increases. Direct transfer and interlingua form the two extremes of this trade-off: interlingua incurs no cost for transfer but the highest cost for analysis and generation, while direct transfer does not incur costs for analysis but poses considerable costs for transfer. Cost here can refer to both computational complexity and difficulty. Accordingly, an approach which has to traverse a long distance for analysis and generation in the triangle may suffer from error propagation.

Direct Transfer

On the lowest level, the level of direct transfer, translation is performed considering only the surface form of the sentence. In rule-based translation on this level, a bilingual dictionary would be used to look up and translate each word in the English sentence. The initial word-by-word translation can then be reordered by simple rules to ensure the correct word order in the target language (Jurafsky and Martin, 2009). The retrieval of individual lexical translations must not be limited to a simple dictionary lookup, but more advanced decision algorithms, similar to the task of word sense disambiguation, can be used (for an example algorithm for translation of "much" and "many" into Russian, see Fig. 25.7, p. 883 of Jurafsky and Martin, 2009). While this method of translation is not in use today, more recent approaches like phrase-based machine translation still follow the basic notion of transforming a source sentence into the target sentence.

Syntactic and Semantic Transfer

The first layer of transfer above direct transfer is syntactic transfer. In this case, the source language string would undergo syntactic analysis — hence the upward arrow labeled "analysis" — and would only then be transferred in its syntactically analyzed form. On the target-language side, the transferred syntactic structure is used to generate target-language words — hence the downward arrow labeled "generation." A more recent example of such a system is the tectogrammatical approach to machine translation (Hajič, 2002) as well as the subsequent uses of abstract meaning representation as a form of semantic representation (AMR, Banarescu et al., 2013). Fundamental to these approaches is the idea that while languages differ significantly on the surface, their differences are smaller on the syntactic and semantic level and transfer at these levels can thus minimize the distance between the languages (Hajič, 2002). This additionally allows to circumvent the treatment of language-specific phenomena during transfer by delegating them to a separate analysis and generation phase where they can potentially be handled in a simpler and more principled manner.

The motivation behind this approach can be best illustrated on examples from language pairs with significant differences. Figure 2.2 shows an English sentence with two semantically equivalent translations into German. The two translations differ mostly in word order: The first example employs the auxiliary verb "hat" combined with the

finite verb "gesehen," thus requiring the specific word order in this example. The second example uses the simple preterite verb "sah," thus following the same word order as the English sentence. Figure 2.3 (bottom half) further shows a simple example of regular word order differences, in this case for the position of adjectives and nouns in English and Spanish: adjectives generally precede nouns in English while they succeed nouns in Spanish.

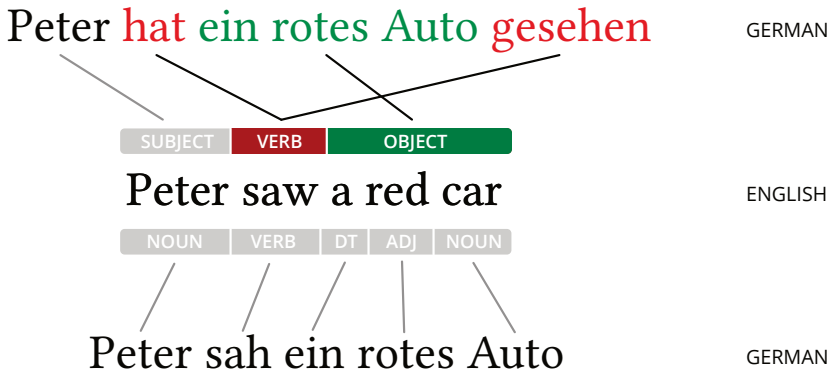


Figure 2.2: Parallel example sentences from English–German.

A further common cross-lingual difference is a language's use of determiners. Figure 2.3 (top half) illustrates this issue on the language pair English–Czech: while the English object contains the article "a," Czech does not use articles and thus has no equivalent token. Figure 2.3 also highlights a third phenomenon whose treatment can be simplified in transfer approaches: morphological agreement. In the graphic, this phenomenon is illustrated by dashed lines between words indicating morphological agreement. There is agreement between the subject and verb as well as within the object noun phrase in both Czech and Spanish. While agreement plays almost no role in morphologically impoverished languages such as English, it is a common and important occurrence in languages such as Czech and Spanish.

Finally, prepositions and the morphological case they co-occur with can pose cross-lingual difficulties. As grammatical case is not expressed morphologically in English, the choice of preposition and the form of the noun phrase appear independent on the surface (e.g. "by *the red car*" vs. "near *the red car*"). In languages such as German, however, the choice of a preposition has to go hand-in-hand with suitable morphological case (e.g. "von *dem roten Auto*" [dative] vs. "nahe *des roten Autos*" [genitive]).

Transfer-based approaches are motivated by the idea that the syntactic or semantic analysis enables a simple treatment of most of these difficult phenomena. For very regular language differences, such as the word order difference between the position of the adjective and noun discussed earlier for English–Spanish, a simple rule would suffice to transform the word order of the source language into the required target-language word order: $\texttt{NP}_{\texttt{English}} \rightarrow \texttt{ADJ NOUN} \Rightarrow \texttt{NP}_{\texttt{Spanish}} \rightarrow \texttt{NOUN ADJ}$.

Beyond simplifying the treatment of some phenomena in classical rule-based ap-

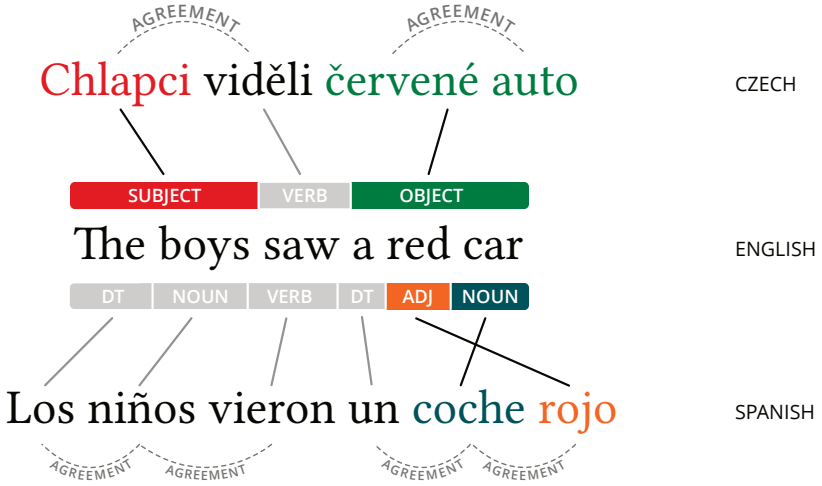


Figure 2.3: Parallel example sentences from English–Czech and English–Spanish.

proaches, the syntactic and semantic abstractions employed also offer useful properties in more modern approaches. Figure 2.4 shows a motivating example from a transfer-based machine translation system for English–Czech (Popel and Žabokrtský, 2010). The example shows the English sentence "Peter does not love Mary" and the Czech equivalent "Petr nemiluje Marii" and highlights three levels of analysis: (1) a morphological level, in which the sentence is divided into the smallest meaning-carrying units, (2) a syntactic layer, in which the syntactic relations between the elements of each sentence are modeled, and (3) a semantic layer (or *tectogrammatic layer* in the terminology of the Prague Dependency Treebank), in which the sentence is reduced to its core meaning. While this semantic layer contains all content words, function words such as articles and prepositions are not nodes in this representation. Note that while the Czech and English sentence structures differ significantly on the syntactic and morphological level, their structures are comparable on the semantic level. The hypothesis underlying these transfer-based systems is that languages are typologically more similar on the tectogrammatical layer. Mareček (2009) explicitly tests this hypothesis for English–Czech and shows that such a "deep" sentence representation can be used to obtain better word alignments than if word alignment is performed on the surface forms only. We will describe machine translation systems based on this idea in more detail in Section 3.1.

Interlingua

The semantic and syntactic transfer approaches discussed above require a separate transfer model for each language pair involved. Assuming that all languages share common properties and that certain parts of language are "universal," it might be possible to produce a semantic representation that fits all languages at the same time. This is the

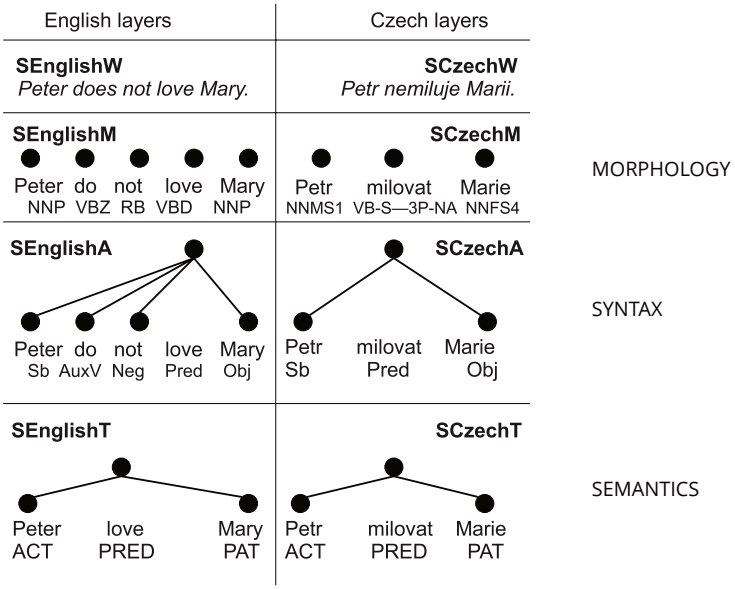


Figure 2.4: Example of English–Czech syntactic and tectogrammatical analysis used in transfer-based machine translation, reprinted from Popel and Žabokrtský (2010).

idea behind *interlingua* (sometimes also referred to as "pivot"): to abstractly represent meaning in a language-independent manner. An interlingua-based machine translation system would simplify dealing with many language pairs at once, since analysis and generation modules could be reused. Interlingua-based approaches saw a significant amount of research (see the overview of interlingua-based approaches in Dorr et al., 1999); however, the idea saw limited practical use. In a similar spirit to interlingua approaches, several modern multilingual neural approaches to machine translation are combining multilingual corpora into a single training set, enabling the translation system a certain amount of generalization (Firat et al., 2016; Johnson et al., 2016).

This section has reviewed how classical approaches to machine translation have modeled the translation process. In the next part, we will introduce phrase-based machine translation models, which have constituted a departure from the more explicitly defined structure of the classical models towards weaker modeling assumptions and greater reliance on parallel data.

2.3 Phrase-Based Machine Translation

Starting with the work performed at IBM Research in the late 1980s and early 1990s (Brown et al., 1988, 1990, 1993), statistical approaches to machine translation demonstrated early successes and became the dominant approach in the field. Although the ideas behind the statistical approach were not new at the time — the general idea was

first proposed by Warren Weaver in 1949 (Weaver, 1955) — the increasing availability of data and better computational resources made it feasible and enabled successful implementations of these ideas.

While the resulting word-based translation models did not find broad adoption by themselves, as word alignment models they provide a fundamental ingredient for phrase-based machine translation systems. In this section, we introduce phrase-based machine translation, starting with an overview of word alignment methods (Section 2.3.1). We will then provide an introduction to model estimation for phrase-based translation models (Section 2.3.2), including phrase extraction and tuning, discuss evaluation of machine translation (Section 2.3.3), decoding strategies (Section 2.3.4) and finally reordering (Section 2.3.5).

2.3.1 Word Alignment

The task of finding alignment links between the words of a bilingual sentence-aligned corpus is a fundamental step in phrase-based machine translation systems. The most commonly used tool for this task, GIZA++ (Och and Ney, 2003), is based on IBM Models 1-5 (Brown et al., 1993).

IBM Models

IBM models assume a statistical view of the language translation process. A sequence of words $\mathbf{s}$ in the source language is translated into a sequence of words $\mathbf{t}$ in the target language. Statistical machine translation assumes that every target-language sequence $\mathbf{t}$ is a possible translation of $\mathbf{s}$ and that a probability $P(\mathbf{t} \mid \mathbf{s})$ for the translation, i.e. for the pair $(\mathbf{s}, \mathbf{t})$, can be assigned (Brown et al., 1993). Using Bayes' theorem, the initial formulation of the task is:

$$P(\mathbf{t} \mid \mathbf{s}) = \frac{P(\mathbf{s} \mid \mathbf{t})P(\mathbf{t})}{P(\mathbf{s})} \tag{2.1}$$

The goal of the prediction task is to find the sequence $\hat{\mathbf{t}}$ that is the sequence produced from $\mathbf{s}$ with the highest probability. Hence:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{s} \mid \mathbf{t})P(\mathbf{t}) \tag{2.2}$$

This is the most fundamental equation of statistical machine translation; it highlights three important components of most statistical machine translation systems: (a) the translation model $P(\mathbf{s} \mid \mathbf{t})$, (b) the language model $P(\mathbf{t})$, and (c) the decoding/search strategy $\arg\max_{\mathbf{t}}$ for finding the most likely sequence $\hat{\mathbf{t}}$.

The IBM models are a series of probabilistic translation models estimated by maximum likelihood using the Expectation Maximization algorithm (Baum, 1972). Fundamental to all IBM models is the notion of an alignment between a pair of strings.

Alignments are introduced in order to factorize the joint likelihood into simpler components. An alignment indicates for each word of the source sentence, by which words in the target-language sentence it was produced. IBM models increase in complexity with every model and the parameters of each model are used to initialize the following model. Model 1 and 2 are the most basic models, allowing exact computation of the Expectation Maximization algorithm. The models marginalize over the alignments **a** between **t** and **s**. The likelihood for $(\mathbf{s} \mid \mathbf{t})$ in terms of $P(\mathbf{s}, \mathbf{a} \mid \mathbf{t})$ is

$$P(\mathbf{s} \mid \mathbf{t}) = \sum_{\mathbf{a}} P(\mathbf{s}, \mathbf{a} \mid \mathbf{t}) \tag{2.3}$$

and $P(\mathbf{s}, \mathbf{a} \mid \mathbf{t})$ can be defined as follows:

$$P(\mathbf{s}, \mathbf{a} \mid \mathbf{t}) = P(m \mid \mathbf{t}) \prod_{j=1}^{m} P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, \mathbf{t}) P(s_j \mid a_1^{j}, s_1^{j-1}, m, \mathbf{t}) \tag{2.4}$$

where the source string $\mathbf{s} = s_1^m \equiv s_1 s_2 ... s_m$ has m words, the target string $\mathbf{t} = t_1^l \equiv t_1 t_2 ... t_l$ has l words, and the alignment **a** is represented by a vector $a_1^m \equiv a_1 a_2 ... a_m$ of m values between 0 and l. If there is an alignment link between the word in position j of the source string and the word in position i of the target string, then $a_j = i$ and if the word is not connected to any target word, then $a_j = 0$.

IBM Model 1 starts with strong independence assumptions that make the model tractable. The first assumption is that $P(m \mid \mathbf{t})$ is independent of **t** and m, meaning that the length of the source sentence is chosen uniformly, regardless of the target sentence. The second assumption is that $P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, \mathbf{t})$ depends only on l, the length of **t**. Finally, $P(s_j \mid a_1^j, s_1^{j-1}, m, \mathbf{t})$ is assumed to only depend on s_j and t_{a_j}.

Thus, the joint likelihood for IBM Model 1 reduces to:

$$P(\mathbf{s}, \mathbf{a} \mid \mathbf{t}) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(s_j \mid t_{a_j}), \tag{2.5}$$

where $t(s_j \mid t_{a_j}) \equiv P(s_j \mid a_1^j, s_1^{j-1}, m, \mathbf{t})$, which is also called the *translation probability* of s_j given t_{a_j}, and ϵ is a small fixed number. Due to the assumptions made in the model, $P(\mathbf{s} \mid \mathbf{t})$ reduces to the following:[2]

$$P(\mathbf{s} \mid \mathbf{t}) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(s_j \mid t_i). \tag{2.6}$$

This form of the final model makes it straight-forward to estimate the parameters $t(s \mid t)$ using Expectation Maximization and since $P(\mathbf{s} \mid \mathbf{t})$ has a unique local maximum for Model 1 (cf. Appedix B of Brown et al., 1993), its parameters can be initialized uniformly.

[2]For the sake of brevity, we avoid the full derivations here but refer the interested reader to Brown et al. (1993).

IBM Model 1 makes no assumptions about the positions of words in either the target or source sentence. While this enables easier estimation, it is an unrealistic assumption. IBM Model 2 thus assumes that $P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, \mathbf{t})$ depends on j, a_j, m and l, while keeping all other assumptions of Model 1. For this, *alignment probabilities* $a(a_j \mid j, m, l)$ are added:

$$a(a_j \mid j, m, l) \equiv P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, l), \tag{2.7}$$

such that

$$\sum_{i=0}^{l} a(i \mid j, m, l) = 1. \tag{2.8}$$

The final form of IBM Model 2 is:

$$P(\mathbf{s} \mid \mathbf{t}) = \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(s_j \mid t_i) a(i \mid j, m, l). \tag{2.9}$$

IBM Model 1 can be seen as a special case of IBM Model 2, where $a(i \mid j, m, l) = (l+1)^{-1}$. Hence, parameters from IBM Model 1 can be reused as parameters for IBM Model 2.

IBM Model 3-5 are significantly more complex than IBM Model 1 and 2 and add several new concepts. *Fertility* describes the number of source words a word is aligned to. The fertility ϕ_i of word t_i in position i is defined as

$$\phi_i = \sum_j \delta(a_j, i), \tag{2.10}$$

where δ is the Kronecker delta function which equals 1 if its arguments are the same and 0 otherwise (Och and Ney, 2003). The possibly empty list of source words aligned to each of the words in the target sentence $\mathbf{t}$ is called a *tablet*. The set of tablets for $\mathbf{t}$ is called a *tableau* of $\mathbf{t}$. The tableau can be thought of as a segmentation of the source and target sentence into aligned units, similar to the phrases we will discuss in the next section. The generative process first chooses a tableau and then reorders its elements to produce $\mathbf{s}$. The resulting permutation is the random variable Π and Π_{ik} is the random variable for the position in $\mathbf{s}$ of word k in tablet i. The joint likelihood for a tableau τ and a permutation π is defined as:

$$
\begin{aligned}
P(\tau, \pi \mid \mathbf{t}) = & \prod_{i=1}^{l} P(\phi_i \mid \phi_1^{i-1}, \mathbf{t}) P(\phi_0 \mid \phi_1^{l}, \mathbf{t}) \times \\
& \prod_{i=0}^{l} \prod_{k=1}^{\phi_i} P(\tau_{ik} \mid \tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^{l}, \mathbf{t}) \times \\
& \prod_{i=1}^{l} \prod_{k=1}^{\phi_i} P(\pi_{ik} \mid \pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^{l}, \phi_0^{l}, \mathbf{t}) \times \\
& \prod_{k=1}^{\phi_0} P(\pi_{0k} \mid \pi_{01}^{k-1}, \pi_1^{l}, \tau_0^{l}, \phi_0^{l}, \mathbf{t}),
\end{aligned}
\tag{2.11}
$$

where τ_{i1}^{k-1} is the series $\tau_{i1}, ..., \tau_{ik-1}$, π_{i1}^{k-1} is $\pi_{i1}, ..., \pi_{ik-1}$ and ϕ_i represents ϕ_{t_i}. Several combinations of ϕ and π may lead to the same $\mathbf{s}$ and $\mathbf{a}$, hence the likelihood is defined over the set of all such pairs $\langle \mathbf{s}, \mathbf{a} \rangle$:

$$
P(\mathbf{s}, \mathbf{a} \mid \mathbf{t}) = \sum_{(\tau,\pi) \in \langle \mathbf{s}, \mathbf{a} \rangle} P(\tau, \pi \mid \mathbf{t}) \tag{2.12}
$$

Model 3 is the first model with fertility. It defines $P(\phi_i \mid \phi_1^{i-1}, \mathbf{t})$ to depend only on ϕ_i and t_i, defines $P(\tau_{ik} \mid \tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^{l}, \mathbf{t})$ to depend on τ_{ik} and t_i, and defines $P(\pi_{ik} \mid \pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^{l}, \phi_0^{l}, \mathbf{t})$ to depend on π_{ik}, i, m and l. Overall, the model estimates three sets of parameters: *translation probabilities*, *fertility probabilities*, and *distortion probabilities*.

Model 4 is based on the intuition that words in the target sentence form larger units (phrases) that often move together. Two new concepts are introduced: $\odot_i$ is the ceiling of the average positions of the source-language words in a tablet, and the *head* of a tablet is the word with the smallest position in the source string. Model 4 replaces $d(j \mid i, m, l)$ with two sets of parameters: one for placing the head and one for placing the remaining words. For placing the head ($\tau_{[i]1}$):

$$
P(\Pi_{[i]1} = j \mid \pi_1^{[i]-1}, \tau_0^{l}, \phi_0^{l}, \mathbf{t}) = d_1(j - \odot_{i-1} \mid \mathcal{A}(t_{[i-1]}), \mathcal{B}(s_j)), \tag{2.13}
$$

where $\mathcal{A}$ and $\mathcal{B}$ map source and target words into word classes (Brown et al., 1990, 1992). And for all other words:

$$
P(\Pi_{[i]k} = j \mid \pi_{[i]1}^{k-1}, \pi_1^{[i]-1}, \tau_0^{l}, \phi_0^{l}, \mathbf{t}) = d_{>1}(j - \pi_{[i]k-1} \mid \mathcal{B}(s_j)) \tag{2.14}
$$

For both Model 3 and 4, the formulation of the models lead to the problem of *deficiency* (Brown et al., 1993): not all of the probability mass is concentrated on events of interest. This is the case since the model can waste probability on strings in which some positions are connected to multiple words and some to none. IBM Model 5 is a reformulation of Model 4, in which the alignment model avoids this deficiency (Och and Ney, 2003).

Word Alignment with IBM Models

While modeling the probability of a sentence pair, the IBM models establish word alignments. To use the models in the word alignment task, the most probable alignment for each sentence pair is determined (*Viterbi alignment*). This is done as a final step after training the models with the Expectation Maximization algorithm over the full dataset. One peculiarity of the formulation of the IBM models is that they generate source-language words from aligned target-language words, which implies that each source-language word is aligned to at most one target word. Since this restriction produces unrealistic word alignments, word alignment models are run in both directions and then combined using various heuristics. This process is commonly referred to as symmetrization and was first introduced by Och and Ney (2003). For a more detailed treatment of symmetrization, we refer the reader to Koehn (2010).

Other Commonly Used Word Alignment Techniques

While word alignment based on IBM models has constituted a core method in statistical machine translation, a number of approaches have been proposed since then. We will only highlight two approaches relevant to this thesis here.

Vogel et al. (1996) presents a simple word alignment model in which the alignment probabilities depend on the differences in the alignment positions in the form of a first-order Hidden Markov model. This model is motivated by the fact that in word alignments there is often a strong dependence of an alignment link a_j on the previous alignment a_{j-1}. Therefore, the model introduces this dependence as

$$P(a_j \mid a_{j-1}, I), \tag{2.15}$$

where I is the length of the source sentence.

A second commonly used word alignment technique was introduced by Dyer et al. (2013). Their method, commonly referred to as *fast_align*, is a log-linear reparametrization of IBM Model 2 that performs as well as IBM Model 4 while being an order of magnitude faster.

2.3.2 Phrase-Based Models

Phrase-based models translate based on short sequences of words (*phrases*). The phrases used in phrase-based machine translation systems are not linguistically motivated but are determined by word alignments. Phrase-based models follow the same basic formulation that we have seen for word-based translation models above in Equation 2.2:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{s} \mid \mathbf{t}) P(\mathbf{t}), \tag{2.16}$$

where $P(\mathbf{s} \mid \mathbf{t})$ is the translation model and $P(\mathbf{t})$ is the language model.

In a basic phrase-based translation system $P(\mathbf{s} \mid \mathbf{t})$ is decomposed into its phrases:

$$P(\bar{s}_1^I \mid \bar{t}_1^I) = \prod_{i=1}^{I} \phi(\bar{s}_i|\bar{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1), \tag{2.17}$$

where the source sentence $\mathbf{s}$ consists of I phrases $\bar{s}_i$, $\phi(\bar{s}_i|\bar{t}_i)$ is the phrase translation probability, $d(\text{start}_i - \text{end}_{i-1} - 1)$ is a distance-based reordering model, start_i the position of the first word of phrase i and end_{i-1} is the position of the last word of phrase $i-1$. In this formulation, segmentation is not modeled in an explicit fashion and we assume that all segmentations are equally likely. Apart from these basic components, a number of other functions such as the reverse translation probability $\phi(\bar{t}_i|\bar{s}_i)$ have empirically proven useful in translation. Hence, to produce a more general model and to assign weights to each of the components, phrase-based models are usually formulated as a log-linear model. In these models, the probabilistic formulation used in the models introduced so far is dropped in favor of a formulation that optimizes towards an evaluation metric. Optimizing towards a metric such as BLEU turned out to be a better proxy for translation quality than likelihood for phrase-based models. It also increases the flexibility of the model since optimizing towards a metric lifts the requirements of a fully probabilistic treatment, thus avoiding potentially intractable marginalizations. We will discuss such metrics and the optimization process in more detail in Section 2.3.3. The log-linear model formulation is:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \exp \sum_{i=1}^{n} \lambda_i h_i(x), \tag{2.18}$$

where $x = (\mathbf{t}, \mathbf{s})$, λ_i is the weight for feature i, and $h_i(x)$ is the ith feature function. Commonly used feature functions include:

- Bidirectional phrase translation probabilities.
- Lexical weighting. Lexical translation probabilities act as a type of smoothing for the phrase translation probabilities by providing an estimate for the lexical translation probability based on word translation probabilities of the words in the phrase. The feature function is defined as (following the formulation used in Koehn, 2010):

$$\text{lex}(\bar{t} \mid \bar{s}, a) = \prod_{i=1}^{\text{len}(\bar{t})} \frac{1}{|\{j \mid (i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(t_i \mid s_i), \tag{2.19}$$

  where $w(t_i \mid s_i)$ is the lexical translation probability for the words t_i and s_i, which is estimated by maximum likelihood from the word-aligned parallel corpus.
- Word and phrase penalties. The word and phrase penalties score the number of phrases and words produced and can encourage the model to produce more or fewer phrases and words.
- n-gram language model.
- Distortion-based reordering model (cf. Section 2.3.5).

Phrase Extraction

The atomic unit of phrase-based machine translation models is the phrase. Accordingly, the quality of the phrase table is an important factor in the overall translation quality. Various methods for extracting phrases from a parallel corpus have been proposed; in the most common approach, phrases are extracted directly from word alignments. To extract phrases from word alignments, sequences of words that are consistent with the word alignment are extracted from the parallel sentences. A phrase pair $(\bar{s}, \bar{t})$ is consistent with an alignment A (following the definition in Koehn, 2010) if no word in $\bar{s}$ has an alignment to a word outside of $\bar{t}$ and no word in $\bar{t}$ has an alignment to a word outside of $\bar{s}$. Formally, a phrase pair $(\bar{s}, \bar{t})$ is consistent with A if and only if:

$$\begin{aligned} &\forall t_i \in \bar{t} : (t_i, s_j) \in A \Rightarrow s_j \in \bar{s} \\ \text{and } &\forall s_j \in \bar{s} : (t_i, s_j) \in A \Rightarrow t_i \in \bar{t} \\ \text{and } &\exists t_i \in \bar{t}, s_j \in \bar{s} : (t_i, s_j) \in A \end{aligned} \tag{2.20}$$

Crucially, this definition of phrase pairs entails that phrases have to be continuous, i.e. without any gaps. The example sentence presented in Figure 2.2, "Peter saw a red car" / "Peter hat ein rotes Auto gesehen," illustrates why this definition can be problematic. In this example, the English word "saw" is aligned to the German discontinuous phrase "hat ... gesehen." In a standard phrase-based machine translation system this would not be a valid phrase pair. While relaxing this constraint is desirable from a linguistic point of view and would improve modeling, it raises computational issues for phrase extraction and decoding.

Chiang (2005) introduces a model using hierarchical phrases. Hierarchical phrase-based machine translation uses a synchronous context-free grammar learnt from parallel sentences. This enables the use of more expressive phrase pairs and improves translation quality at the expense of increased computational complexity: decoding without a language model in this model can be performed using the $O(n^3)$ CKY algorithm (Cocke and Schwartz, 1970; Kasami, 1965; Younger, 1967). Simard et al. (2005) and Galley and Manning (2010) introduce non-hierarchical phrase-based systems allowing non-continuous phrases. Galley and Manning (2010) use suffix arrays to find and represent discontinuous phrases (Lopez, 2007) and extend a multi-stack decoder (Koehn et al., 2007) to be able to handle phrases with gaps efficiently.

2.3.3 Parameter Tuning and Evaluation

As we have introduced them above, phrase-based models contain a set of weights for their individual components. In this section, we will give a brief overview over two issues arising from this: (1) What loss function should the weights be optimized for? (2) What optimization algorithm can be used to determine the weights? The first question goes hand in hand with the question how machine translation systems are evaluated and we will therefore answer this question first.

Evaluation

The question of how to evaluate machine translation systems has been at the core of an active field of machine translation research. The two fundamental areas of interest in this field are *manual* evaluation and *automatic* evaluation. Manual evaluation of machine translation systems is generally performed by bilingual human judges. Such judges assess the quality of translations on several dimensions, most notably the criteria of *fluency* and *adequacy*. Fluency determines whether the output sentence is fluent, including whether there are grammatical errors, problems in morphological agreement, word order errors etc. Adequacy addresses whether the output sentence conveys the same meaning as the input sentence, hence whether it is an *adequate*/acceptable translation of the input sentence. As manual evaluation uses human judgements directly, it is undoubtedly desirable; however, manual evaluation also incurs significant costs: it is slow and expensive.

Automatic evaluation is the attempt to approximate human judgement of translation quality automatically. In their most common form, automatic evaluation metrics rely on having access to one or several *reference translations*, i.e. translations of the input sentence produced by professional translators. Based on the source sentence and the reference translations, an evaluation metric will assign a score to the output of a machine translation system (translation hypothesis). Automatic evaluation metrics themselves can be evaluated by determining their correlation to human judgements, as is done in the annually organized WMT metrics task (Bojar et al., 2016). In such tasks, evaluation metrics are evaluated in two categories: metrics are evaluated for system-level (or corpus-level) correlation, which measures how well a metric's scores correlate with the ranking of the translation systems (i.e., over the full evaluation corpus), and segment-level (or sentence-level) correlation, which measures how well a metric performs in judging translations of a specific sentence. Since automatic metrics allow for fast and non-interactive evaluation, they are also commonly used to optimize the parameters of machine translation systems.

BLEU While a fairly simple evaluation metric, BLEU (Papineni et al., 2002), short for Bilingual Evaluation Understudy, has become the most popular metric in recent years. This is partially due to its simplicity and due to its selection in important evaluation campaigns such as WMT and evaluation campaigns organized by DARPA and NIST. However, BLEU has also consistently shown high correlation with human judgements on the corpus level. BLEU is based on n-gram match between the hypothesis translation and the reference translations. The metric is defined as follows:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{2.21}$$

where p_n is the precision for n-grams of length n and w_n is a positive weight such that $\sum_{n=1}^{N} w_n = 1$. n-gram precision is defined with the sentence as the basic unit of

evaluation:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{ngram \in C} \text{count}_{\text{clip}}(ngram)}{\sum_{C' \in \{Candidates\}} \sum_{ngram' \in C'} \text{count}(ngram')}, \tag{2.22}$$

where count($ngram$) indicates how many times an n-gram that occurs in one of the references occurs in the translation hypothesis. To avoid rewarding machine translation systems for overgeneration, $\text{count}_{\text{clip}}(ngram)$ limits how often each hypothesis n-gram match is counted to the maximum number of times the n-gram occurs in any single reference translation. BP is a brevity penalty discouraging short translations, which would otherwise be rewarded under a purely precision-based metric. For a total candidate translation corpus length c (i.e. c is the combined length of all hypothesis translations) and an effective reference corpus length r (r is the sum of the lengths of each reference translation; if there are multiple references, the length closest to the hypothesis translation length is chosen), BP is defined as follows:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}. \tag{2.23}$$

The maximum n-gram length N is commonly set to 4 and the weights are generally uniform with $w_n = \frac{1}{N}$.

Despite its relatively high correlation with human judgements, BLEU has a number of shortcomings and researchers have repeatedly urged the community not to be overly reliant on it (see e.g. Callison-Burch et al., 2006). Frequently raised issues include that beyond allowing multiple reference translations, BLEU does not take into account synonyms (e.g. "town" and "city" are both reasonable translations of the German word "Stadt," but each is only judged as correct if it explicitly appears in the reference translation). Since the metric is based only on n-gram match, it also treats all words as equally important, which can be a problematic assumption (e.g. whether the word "not" appears in a sentence may fundamentally change its meaning). The maximum length of the n-grams considered and the sparsity of potentially longer n-grams also mean that the metric is very local and can only partially account for word order. BLEU scores depend on many factors and are not directly comparable across languages or even accross datasets in the same language. Finally, n-gram scores are always computed over tokens, which is a reasonable abstraction for morphologically impoverished languages such as English, but raises issues for morphologically rich languages in which only matching full tokens can lead to sparsity. This issue is partially addressed by the recent trend of evaluation metrics using subword representations, such as character n-grams in BEER (Stanojević and Sima'an, 2014) or character-based edit distance in CharacTER (Wang et al., 2016).

METEOR METEOR (Banerjee and Lavie, 2005; Denkowski and Lavie, 2014) is an evaluation metric that, unlike BLEU, focuses less on precision and introduces several further improvements. The metric employs paraphrasing tables to expand words with

their synonyms, therefore allowing semantic matches instead of only lexical matches between a translation and its reference. Word order is evaluated over the entire sentence by calculating a fuzzy reordering score based on automatic word alignments between the translation hypothesis and the reference translation. The metric also distinguishes function words and content words, following the intuition that content words are more important in the evaluation. The individual components of the metric are combined as a weighted precision and recall measure and all weights and parameters are optimized based on human judgements from existing evaluation campaigns.

Tuning

The parameters of machine translation systems are optimized to maximize an evaluation metric, most commonly BLEU. In phrase-based machine translation systems, phrases are extracted from a large number of parallel sentences. Since there are only a limited number of components in phrase-based machine translation systems, their weights can be estimated based on a smaller subset of holdout parallel sentences. In the standard experimental setup, the bilingual data is separated into a training set, a development set and a test set and language models are estimated from a separate monolingual corpus. Since BLEU is a corpus-level metric, this estimation cannot be performed on a per-sentence basis.

Och (2003) first proposed an iterative tuning algorithm commonly referred to as MERT (minimum error rate training), which is still one of the most widely used optimization procedures in (non-neural) statistical machine translation systems. MERT directly minimizes the error function (usually BLEU). Since BLEU is not continuously differentiable, MERT uses an alternative optimization method based on iterative line search (similar to Powell search, Och, 2003). Apart from the classic MERT approach, several learning algorithms capable of handling a larger number of features have been proposed. Pairwise ranking approaches (Hopkins and May, 2011) cast tuning as a classification task. These approaches are easy to implement in most phrase-based systems, since they can be implemented in a similar setting as MERT. Various online learning approaches have also shown promising results, especially for large numbers of features (Liang et al., 2006; Watanabe et al., 2007). Cherry and Foster (2012) proposed a batch version of the widely-used MIRA algorithm (Crammer et al., 2006) that performs on par with online algorithms and MERT while at the same time reducing training time and implementation complexity. For a more extensive treatment of optimization techniques in machine translation, we refer the reader to Neubig and Watanabe (2016).

2.3.4 Decoding

Decoding is the process of finding the best-scoring translation according to the model. Even in word-based models, this is a difficult problem as there is a factorial number of possible permutations of words, for each of which a suitable translation has to be selected. The problem has been shown to be NP-complete by reduction from the Hamil-

ton Circuit Problem and the Minimum Set Cover Problem (Knight, 1999). Similarly, phrase-based machine translation can be shown to be NP-complete by reduction from the Traveling Salesman Problem (Zaslavskiy et al., 2009). Since exhaustive search is thus infeasible, a number of heuristic search algorithms are commonly used in phrase-based decoders. The most popular decoding algorithm for phrase-based models is a variant of beam search (Koehn, 2003), which is similar to the decoding strategy proposed for alignment template-based machine translation (Och, 2002; Och and Ney, 2004) and related to the strategy proposed by Tillmann and Ney (2003).

To generate the search space for translation, the input string is matched against the phrase table, generating a set of translation options. The search then proceeds from left to right: starting with an empty sentence, partial translations (hypotheses) are generated by expanding a prior hypothesis until the hypothesis covers the entire sentence. Each completed hypothesis forms a leaf in the search graph and the task of the search algorithm is to find the best-scoring leaf in this graph. Several tricks are applied to lower the amount of computation required. Hypothesis recombination combines different paths in the graph leading to the same hypothesis, thus reducing the search space. This step can be performed in a lossless fashion if two hypotheses cover the same source words and if the last $n - 1$ target words are identical (assuming an n-gram language model). In first-best decoding, hypotheses are recombined by selecting only the better-scoring hypothesis. Apart from reducing the search graph, this technique also helps in handling of spurious ambiguity, i.e., cases in which hypotheses only differ in their segmentation.

Stack decoding is a technique for reducing the search space while minimizing the risk of removing good hypotheses. Pruning the search space of lower-scoring candidate hypotheses poses a risk: low-scoring hypotheses covering only few words may still lead to good translations once fully expanded while high-scoring hypotheses covering many words may lead to bad translations. Both should not be discarded or kept based on their score alone. A solution first introduced by Tillmann et al. (1997) for monotone decoding in word-based models and extended for phrase-based models by Koehn (2004b) is the use of hypothesis stacks. In this method, hypotheses are placed on several stacks based on a specific criteria, e.g. based on the number of source words covered, and pruning is performed on these individual stacks instead of globally. Search in the decoder is then implemented by keeping a beam of most promising hypotheses in each stack while searching for the best complete hypothesis.

Because subsets of the source sentence with the same length may lead to full translations with varying difficulty, only comparing them based on their model score is not sufficient. To address this issue, an estimate of future cost is added to the scoring function (Koehn, 2004b). Future cost provides an estimate of how difficult it will be to translate the rest of the sentence, hence discouraging the pruning strategy from giving too much preference to hypotheses covering only easy parts of the sentence. Calculating the full future model score for every hypothesis would be computationally infeasible; hence, when estimating this future cost (sometimes also called outside cost), phrase-based decoders only consider translation model and local language model scores while ignoring reordering and language model interactions.

2.3.5 Reordering

One of the issues that make phrase-based machine translation computationally expensive and more difficult to model is reordering. Depending on the language pair, phrases may have to be reordered significantly to arrive at a good translation. Without stack pruning, the decoding task would be exponential. For unrestricted reordering, the computational complexity of decoding with stack pruning is (Koehn, 2010):

$$O(\text{max stack size} \times \text{sentence length}^2) \tag{2.24}$$

Reordering is generally restricted using a reordering limit d. In this case, phrases can only skip a maximum of d words if they are reordered. Using a constant reordering limit, the complexity of decoding can be reduced to:

$$O(\text{max stack size} \times \text{sentence length}) \tag{2.25}$$

Apart from a simple word-based reordering limit d, several alternative reordering constraints, such as constraints based on Inversion Transduction Grammar (ITG, Wu, 1997) can be employed. Zens and Ney (2003) formalize various reordering constraints used in phrase-based systems and evaluate whether basic ITG constraints are sufficient for English–German and English–French translation. Lopez (2009) further provides a formalization of statistical machine translation as weighted deduction, which includes a formal specification of various reordering constraints.

Two reordering models are most commonly used in phrase-based machine translation: distance-based reordering models and lexicalized reordering models. The distance-based reordering model provides a simple measure for distortion during decoding. In the initial formulation of phrase-based machine translation introduced above (again assuming all segmentations are equally likely), this model was represented as $d(\text{start}_i - \text{end}_{i-1} - 1)$:

$$P(\bar{s}_1^I \mid \bar{t}_1^I) = \prod_{i=1}^{I} \phi(\bar{s}_i|\bar{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1), \tag{2.26}$$

where the source sentence $\mathbf{s}$ consists of I phrases $\bar{s}_i$, start_i is the position of the first word of phrase i and end_{i-1} is the position of the last word of phrase $i-1$. The distance captured by this model is the number of words skipped when source words are out of sequence. For two phrases in sequence, the distance is $d(0)$. $d(x)$ is then defined as an exponentially decaying function $d(x) = \alpha^{|x|}$, where $\alpha \in [0, 1]$, thus imposing a high cost on long movements and a low cost on shorter movements.

Unlike the basic distance-based reordering model, which only conditions on the distance, lexicalized reordering models also rely on the phrase itself. These models are also often called MSD models, a name derived from the three types of movements they allow: two phrases can be in monotone order (M), a phrase can be swapped with the previous phrase (S), and two phrases can be discontinuous (D). The reordering model

estimates the probability of a sequence of orientations $\mathbf{o} \equiv o_1 o_2 ... o_n$:[3]

$$P(\mathbf{o} \mid \mathbf{t}, \mathbf{s}) = \prod_{i=1}^{n} P(o_i \mid \bar{t}_i, \bar{s}_{a_i}, a_{i-1}, a_i), \tag{2.27}$$

where $o_i \in \{M, S, D\}$ and a_i, a_{i-1} are the phrase alignments for the current and previous phrase. Lexicalized reordering models were first introduced by Tillman (2004) and extended by Galley and Manning (2008) to add hierarchical phrase reordering.

2.3.6 Preordering

Reordering contributes significantly to the computational complexity of decoding. Disconnecting reordering from translation decisions can therefore lower the cost of decoding significantly. As well as performing it during decoding, reordering can also be approached as a post-processing or pre-processing task. Reordering as pre-processing, also known as preordering or source-side reordering, has proven an effective method to reduce the computational complexity of decoding and to improve the handling of difficult long-distance reordering phenomena. We will discuss various approaches to preordering in more detail in Chapter 4.

2.4 Neural Machine Translation

Recently, neural approaches to machine translation have enjoyed great popularity and success for a broad range of language pairs. Neural machine translation aims to model the full translation process using a single neural network model.

A popular approach to modeling the task is the encoder-decoder framework, in which the source sentence is encoded into a vector, commonly called context vector, and a decoder generates a translation from this vector. A key ingredient to the success of this approach, the attention mechanism, was introduced by Bahdanau et al. (2015). Attention predicts which source tokens will be relevant to the prediction of a given target token.

Specifically, the decoder defines a probability over the translation $\mathbf{t}$ by decomposing it into conditionals:

$$P(\mathbf{t}) = \prod_{i=1}^{l} P(t_i \mid t_1^{i-1}, \mathbf{s}), \tag{2.28}$$

where $\mathbf{s}$ is usually represented as a context vector c and $\mathbf{t} = t_1^l \equiv t_1 t_2 ... t_l$. In the recurrent neural network approach of Bahdanau et al. (2015), the decoder is defined as:

$$P(t_i \mid t_1^{i-1}, \mathbf{s}) = g(t_{i-1}, r_i, c_i), \tag{2.29}$$

[3] We use the formulation from Galley and Manning (2008).

where g is a non-linear function which outputs the probability of t_i, r_i is the hidden state[4] of a recurrent neural network (RNN) at time step i and c_i is the context vector for target word t_i. r_i is defined as:

$$r_i = f(r_{i-1}, y_{i-1}, c_i). \tag{2.30}$$

The context vector c_i is computed using so-called annotations, which are a context representation aiming to summarize the entire source sentence with a focus around a given word. The annotations are produced by running a bidirectional recurrent neural network (Schuster and Paliwal, 1997) in both directions for each token, producing annotations $(h_1, ..., h_m)$ for the m source words.

The context vector c_i is then computed as the weighted sum of the annotations:

$$c_i = \sum_{j=1}^{m} \alpha_{ij} h_j, \tag{2.31}$$

where the weights α_{ij} are computed as the softmax over an alignment model e_{ij}:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{ik})} \tag{2.32}$$

$$e_{ij} = a(r_{i-1}, h_j) \tag{2.33}$$

The alignment model e_{ij} scores how well words around the source position j fit the words around the target position i. This model is usually implemented as a feed-forward neural network, but other approaches have also been proposed (Luong et al., 2015).

Given this model formulation, stochastic gradient descent can be used to train a translation model from parallel data and the resulting model can be used to generate translations via beam search. One limitation of neural machine translation is that the model formulation outlined above relies on a closed vocabulary. Sennrich et al. (2016) as well as Wu et al. (2016) approach this problem by learning a vocabulary of subword units, which can be combined into words and thus allow for open vocabulary translation. Finally, convolutional neural networks have become increasingly popular as an alternative to recurrent neural network-based approaches (e.g., Gehring et al., 2017).

Outlook

In this chapter, we have provided a brief overview of three important approaches to machine translation: classical approaches, phrase-based and neural machine translation. In the next chapter, we will focus on the role and treatment of linguistic structure in these approaches and provide relevant background on the area of linguistic typology.

[4] We use r_i instead of the usual s_i here to avoid confusion with our notation of the source sentence.

Chapter 3

Linguistic Structure in Machine Translation

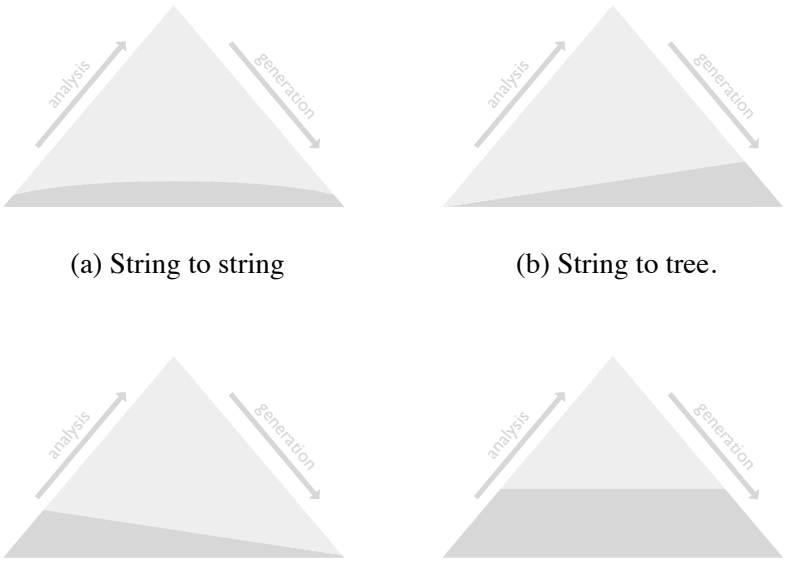
In this chapter, we provide an overview of the role and the treatment of linguistic structure in modern approaches to machine translation and introduce the field of linguistic typology and its role and use in natural language processing.

3.1 Structure in Machine Translation

As a first step, we will summarize how research in machine translation has exploited linguistic and non-linguistic structure in various approaches. These approaches can be categorized into four main areas. String-to-string systems are the most popular architecture and include standard phrase-based machine translation systems as well as many neural approaches to machine translation. Tree-to-string and string-to-tree systems assume some extent of hierarchical structure on either the source or the target side and include most syntax-based machine translation systems. Finally, tree-to-tree systems assume a hierarchical representation for both the source and the target side. This category includes transfer-based approaches such as the tectogrammatical translation approach.

String to String

The most common architecture for machine translation systems assumes no explicit linguistic structure on both the source and target side. This includes the standard phrase-based approach discussed in Section 2.3 as well as recurrent and convolutional neural machine translation, as discussed in Section 2.4. Hierarchical phrase-based machine translation (Chiang, 2007) is based on synchronous context-free grammars and thus includes hierarchical derivations. Since the grammar is learnt based on parallel data and



(a) String to string

(b) String to tree.

(c) Tree to string.

(d) Tree to tree.

Figure 3.1: Structure in the Vauquois triangle.

without explicit consideration for linguistic structure, it is often considered to be a hierarchical string-to-string system (see e.g. Williams et al., 2016). Hierarchical phrases have also been used to guide the search in attention-based neural machine translation (Stahlberg et al., 2016). While the translation itself may not exploit linguistic hierarchy, it may still be part of the overall translation process. A popular setup is the usage of syntax on the source side to preorder sentences, followed by a purely string-to-string translation step (e.g. Lerner and Petrov, 2013; Khalilov and Sima'an, 2012).

String to Tree and Tree to String

String-to-tree and tree-to-string systems are the two most common forms of syntax-based statistical machine translation. In string-to-tree systems, the derived target-side trees reflect observed linguistic structure. String-to-tree models use synchronous tree substitution grammars (Aho and Ullman, 1972) or synchronous context-free grammars with weakened restrictions. Galley et al. (2006), for example, use rules containing tree fragments only on the target side. The annotation of syntactic treebanks may not always be optimal for use in syntax-based machine translation, hence Huang and Knight (2006) show that relabeling constituents to better fit the translation task can provide significantly better translation performance. Syntax-augmented models (Zollmann et al., 2006; Zollmann and Venugopal, 2006) label phrases with syntactically motivated categories. Syntactically motivated models can improve word order and general translation quality, but their structures can also enable easier treatment of other phenomena. Williams and Koehn (2011) show that morphological agreement can be improved via

unification-based constraints on the target side of a string-to-tree model. Sennrich and Haddow (2015) show that hierarchical structure on the target side can also help with other morphological phenomena, such as particle verbs and composita, when enabling the system to compose subword units.

In tree-to-string translation models, derivations contain linguistic structure on the source side. Neubig and Duh (2014) study the factors influencing the quality of tree-to-string translation systems and find that alignment, parse quality and search strategy are crucial components determining whether a tree-to-string system performs on par with or significantly better than a string-to-string system. Apart from models based on constituency structure, dependency structure has also been employed in both string-to-tree and tree-to-string approaches. Shen et al. (2008) introduce a system producing dependeny graphs on the target side, which enables language models to better capture long-distance relations. Treelet-based approaches (Quirk et al., 2005) employ dependency trees on the source side.

For neural machine translation, Aharoni and Goldberg (2017) show that translating into English represented as linearized, lexical constituency trees can improve translation quality. Their manual evaluation indicates that the observed improvements are partially caused by improved reordering. Eriguchi et al. (2016) introduce a neural translation model with a tree-to-string attention mechanism in which attention can focus on individual words as well as whole phrases on the source side. Finally, Bastings et al. (2017) use graph-convolutional networks to integrate source dependency trees directly into an attention-based model.

Tree to Tree

Žabokrtský et al. (2008), Popel and Žabokrtský (2010) and Dušek et al. (2012) describe a statistical machine translation system based on transfer at the semantic level. The system uses a standard dependency parser to produce the syntactic layer in the source language, which is then followed by a deterministic transformation producing the tectogrammatical layer. Transfer is performed using a translation model based on Maximum Entropy models (Mareček et al., 2010) and a Hidden Markov model defined over edges of the tree (Žabokrtský and Popel, 2009). In the target language, the tectogrammatical representation that contains abstract grammatical information is then deterministically transformed (based on hand-written rules) into surface forms. Empirically, this English–Czech system, as well as systems more recently developed for English–Spanish (Labaka et al., 2015), do not perform as well as phrase-based machine translation systems when measured in terms of BLEU and require significant amounts of language resources and development for each language pair.

3.2 Linguistic Typology

Languages are diverse in structure and the debate on whether true linguistic universals exist and attempts to pinpoint those have been part of linguistic theory for at least the past several decades. The study of linguistic typology itself dates back to as early as the 18th century. Graffi (2010) highlights Adam Smith's "Dissertation on the Origin of Language" (Smith, 1767) as one of the first works in this field. Georg von der Gabelentz's 1894 paper "Hypologie der Sprachen, eine neue Aufgabe der Linguistik" (von der Gabelentz, 1894) is often pointed to as initially introducing the name typology. His broader work is credited with introducing modern concepts of linguistic typology while separating typology from genealogy of language and excluding any subjective assessment of languages in terms of "quality," a common notion until this point. In the modern sense, linguistic typology studies the differences between languages and describes which generalizations can be made about cross-linguistic variation (Daniel, 2010).

3.2.1 Description of Language Universals

Language universals fall into several categories based on the strength of their claim. Moravcsik (2010) divides universals into four categories based on two parameters: unrestricted/restricted universals and absolute/statistical universals. Unrestricted universals can be absolute ("all languages have feature X") or statistical ("most languages have feature X"). Accordingly, restricted universals, also called implicational universals, can be absolute ("for all languages, if a language has feature X, it also has feature Y") and statistical ("for most languages, if a language has feature X, it also has feature Y"). Implicational universals were first introduced by Jakobson (1941), who studied the acquisition of sounds and introduced universals such as "there exists no language which has velar stops without having labial stops" (Graffi, 2010). Greenberg (1966b) further distinguishes a separate category for implications in both directions, i.e. equivalence.

A high correlation between two parameters or even an absolute implication in both directions can indicate that the two seemingly independent parameters may be one broader parameter in language. Daniel (2010) illustrates this phenomenon with the example of noun phrase case marking and word order: that the lack of case marking on noun phrases tends to co-occur with strict word order could be explained by a parameter for the "choice of formal means to mark grammatical relations" (Daniel, 2010).

3.2.2 Language Universals and Generative Grammar

Linguistic typology differs from other approaches involving linguistic universals, such as generative grammar, mainly in the question of the origin and the form of linguistic universals. Universals in linguistic typology are "empirically established generalizations that describe distributional patterns for grammatical phenomena across languages" (Cristofaro, 2010) while in generative grammar they are a "set of entities that

are specifically represented in a speaker's mental grammar" (Cristofaro, 2010). Linguistic typology hence deals with language universals in a descriptive manner. Linguistic universals in generative grammar, on the other hand, are motivated by hypotheses about the restrictions of language acquisition (Daniel, 2010). Accordingly, Daniel (2010) argues, the biggest differences between both approaches to language universals arise from the varying basic assumptions of both approaches: generative grammar posits that all languages are mostly structurally identical while linguistic typology makes no prior assumptions about structure but collects evidence for whether or not they are. This has implications on the methodology of research in both fields. Given the basic premise that all languages are essentially similar and that there exists an innate universal grammar whose parameters are fine-tuned for a specific language, it is reasonable to focus on one or few languages initially. In generative grammar research, this initial focus was mostly on English and was gradually expanded to other languages. Linguistic typology research, lacking this basic assumption, focuses on as broad and diverse a set of languages as possible. Another methodological difference is that the assumptions made by generative grammar legitimize introspection as a valid methodology of data collection while linguistic typology research has relied on corpora and usage-based studies (Daniel, 2010).

3.2.3 Word Order and Morphology

Two of the most central areas of study in linguistic typology are word order and morphology.

Word Order

Word order has been an integral part of typological studies starting with Greenberg (1963), who identified word order patterns based on a sample of 30 languages. His study suggested 45 universals based on the sample, including 25 implicational universals for basic word order. These universals include basic observations that are still considered valid today and have been corroborated by studies involving broader language samples. Greenberg's first universal, reproduced here from Greenberg (1963), deals with the order of subject, verb and object:

Universal 1. In declarative sentences with nominal subject and object, the dominant order is almost always one in which the subject precedes the object.
This leaves us with three common types: VSO, SVO, and SOV.

Examples of statistical and absolute implicational universals for word order are Universal 2 and 25 from Greenberg (1963):

Universal 2. In languages with prepositions, the genitive almost always follows the governing noun, while in languages with postpositions it almost always precedes.

Universal 25. If the pronominal object follows the verb, so does the nominal object.

Greenberg's work was influential and established implicational universals as an important type of language universal, which were subsequently used by many typological studies (Song, 2010). In later research, typologists attempted to reduce Greenberg's universals to more essential principles. Lehmann (1973, 1978) proposed a *Fundamental Principle of Placement*, which reduced many parameters in Greenberg's universals to only one: the order of the verb and the object noun phrase. According to the principle, other word positions follow directly from this parameter: modifiers are placed on the opposite side of V or O, i.e. modifiers go left in OV languages and right in VO languages. Thus, the principle correctly predicts that in a VO language such as French, adjectives would follow the nouns they modify. That some languages do not follow this basic stipulation (e.g. English is a VO language, but adjectives precede nouns) is justified by the claim that such languages are in the process of undergoing a change from OV to VO or vice versa (Lehmann, 1973).

Hawkins (1983) argued against statistical universals and attempted to produce a set of exceptionless universals for word order. He based his studies on a sample of 336 languages. To remove exceptions and the statistical nature of the universals, it is necessary to narrow the scope of the universals drastically. A universal may now have a logical form such as $\mathrm{Pr} \Rightarrow (\mathrm{NA} \Rightarrow \mathrm{NG})$, which expresses that in languages in which adpositions are prepositions, if the adjective follows the noun, genitives also do (Hawkins 1983, as cited in Song, 2010). While this has the potential to increase the number of universals significantly, Hawkins (1983) found that often only a limited set of universals were attested for in the data. For the position of modifiers of nouns in prepositional languages, for example, he found that only seven out of the 32 possible combinations of the positions of demonstratives, numerals, adjectives, relative clauses and genitives in relation to the noun were attested for in the data (Hawkins 1983, as cited in Song, 2010). Instead of based on the position of the verb and the object, the main division of the approach is into prepositional and postpositional languages.

Dryer (1992) introduced the *Branching Direction Theory* (BDT), which returned to OV/VO as the main parameter and all other orientations being implied by it. The BDT predicts that languages tend to be either fully left-branching (phrasal categories precede non-phrasal categories) or fully right-branching (phrasal categories succeed non-phrasal categories) (Dryer 1992, as cited in Song, 2010). The theory introduces a number of modifications to the standard representation to account for exceptions, for example Dryer (1992) argues for treating adjectives and nouns as non-phrasal categories, therefore excluding them from predictions of the BDT, which only predicts the order between phrasal and non-phrasal categories. An important reason for highlighting Dryer's work is that in order to test his hypotheses, he employed the largest sample of languages used until this point, a database consisting of 625 languages and was later one of the co-creators of World Atlas of Language Structures (Dryer and Haspelmath, 2013), a dataset widely used in areas of research beyond linguistic typology itself.

Word Order Freedom

Most word order typologies introduced above describe the relative order of a select set of word types and constituents to each other. Only few language universals in relation to word order freedom have been studied. The most prominent is the hypothesis that languages with more word order freedom have an increased amount of case marking (Kiparsky, 1997; Sapir, 1921). This hypothesis was tested by Futrell et al. (2015) who performed a study of word order freedom in dependency treebanks and found a correlation between word order freedom and the presence of nominative-accusative case marking.

Morphology

In the area of morphology, linguistic typology studies the various systems languages employ to compose words from smaller meaning-carrying units. The traditional typological view of morphology is that there are a small set of "holistic" language types (see Brown, 2010). These types are *inflectional*, *agglutinative* and *isolating* languages. This basic categorization was refined by Sapir (1921) who defined languages along two dimensions. The first dimension, *formal process*, describes the processes a language uses to form words and the second dimension, *synthesis*, describes how many concepts are contained in a word. There are four basic formal processes: In isolating languages, the word expresses the root without any additional morphemes, in agglutinative languages, regular affixes are added to the root, in fusional languages, affixes are added and the root may be changed, and in symbolic languages, changes may be applied to the root itself (see Brown, 2010; Sapir, 1921). Sapir (1921) defined three types of synthesis: analytic, synthetic and polysynthetic, where analytic has the lowest and polysynthetic has the highest morpheme-per-word ratio.

3.2.4 Linguistic Typology and Natural Language Processing

Whether explicitly or implicitly, linguistic typology has long played a role in natural language processing research. Bender (2009) argues that truly language-independent natural language processing requires linguistic knowledge in the form of generalizations from linguistic typology. Linguistic knowledge is often regarded with a critical view in natural language processing research, since attempts to incorporate such knowledge (e.g. in the form of rules) have often had limited success and are difficult to scale across languages and domains. Instead of using linguistic knowledge in the form of rules or hard constraints, Bender (2009) suggests that to build language-independent (instead of "linguistically naïve") natural language processing systems, natural language processing research should be aware and incorporate what is known about the existing human languages instead of attempting to be able to handle every possible (potentially artificial) language.

The fields of linguistic typology, natural language processing and computational linguistics have interacted and benefited from each other in several ways. The World

Atlas of Language Structures Online (WALS, Dryer and Haspelmath, 2013) is the most widely used linguistic typology resource in natural language processing research. Several lines of research have explored how WALS features can be predicted for individual languages. The motivation behind predicting individual features can reach from filling gaps in WALS itself to building language representations for selecting suitable training data (e.g. in delexicalized transfer parsing, McDonald et al., 2011). Rama and Kolachina (2012) and Teh et al. (2007) predict WALS features by building statistical models based on their interactions with other WALS features and Daume III and Campbell (2007) and Lu (2013) attempt to predict typological implications based on WALS.

Other work attempts to predict WALS values from text alone. Östling (2015) uses parallel sentences derived from translations of the Bible to predict WALS word order features, which he then compares to the true values in WALS. Bender (2016) provides an overview of further computational linguistics applications, such as the LinGo Grammar Matrix (Bender et al., 2002), a tool providing a method for bootstrapping a grammar for a new language based on a base grammar and a set of typological parameters.

In natural language processing research, interest in linguistic typology has been driven significantly by the difficulties of adapting existing natural language processing tools to new languages. Most modern natural language processing tools, such as part-of-speech taggers, parsers, etc., require significant amounts of labeled training data, which is not available for many languages. To alleviate this issue, language resources can be artificially created by transferring existing labels available for a high-resource language such as English to the low resource target language. This technique has been applied successfully to part-of-speech tagging, named entity recognition and morphological analysis (Yarowsky et al., 2001), as well as dependency parsing (Hwa et al., 2005). A related approach is delexicalized transfer parsing (Zeman and Resnik, 2008), in which a parser for a target language is trained without words on a related source language or on a mixture of source languages. For both approaches, it is helpful to select source languages sharing typological properties with the target language. Several approaches use WALS to select suitable training data for the target language (Täckström et al., 2013; Søgaard and Wulff, 2012; Naseem et al., 2012). Apart from using it to select training data, typological data has also been used to provide additional guidance to dependency parsers. Aufrant et al. (2016), for example, train delexicalized transfer parsers and remove typological differences in word order by reordering the source language to be closer to the target language using rules derived from WALS.

Finally, in several natural language processing tasks, there are benefits to learning models which are trained on multiple languages at the same time. Apart from being able to bridge the lack of sufficient resources for individual languages, multilingual training data can also be beneficial since structural ambiguities in one language may be explicit in another language and thus considering both languages may help overall (Snyder, 2010; O'Horan et al., 2016). One example of such a model is presented by Ammar et al. (2016), who train a single multilingual model for dependency parsing in several languages. Linguistic typology can both inform the modeling of parameters for the involved language and provide input data for such models.

Outlook

The purpose of the first part of this thesis has been twofold: Firstly, we provided a broad overview of the research field of machine translation, aiming at outlining the three major classes of approaches to machine translation, and therefore providing the necessary context for the empirical results we present in the following chapters. Secondly, we discussed aspects related to the object underlying all machine translation research, namely language itself and its study in the research field of linguistics. We specifically focused on how areas of linguistic study, such as syntax and semantics, are treated in machine translation. Finally, we provided an overview of the research field of linguistic typology, highlighting morphology and word order as two important areas on which we will focus in this thesis.

Part II

Word Order Freedom

The contents of the following three chapters are based on the following publications:

Joachim Daiber and Khalil Sima'an. *Delimiting Morphosyntactic Search Space with Source-Side Reordering Models*. In 1st Deep Machine Translation Workshop, 2015.
Joachim Daiber performed all experiments and wrote the article. Khalil Sima'an provided guidance and helped editing the article. Khalil Sima'an and Joachim Daiber produced the idea for the article.

Joachim Daiber, Miloš Stanojević, Wilker Aziz, and Khalil Sima'an. *Examining the Relationship Between Preordering and Word Order Freedom in Machine Translation*. In First Conference on Machine Translation, 2016.

Miloš Stanojević developed the reordering grammar preordering system, ran preordering experiments for English–Japanese and wrote an initial draft of Section 2.4.2. Wilker Aziz ran translation experiments for English–Japanese, provided the version of Moses with extended lattice support and wrote an initial draft of Section 2.2.3 and Section 2.5.1. Khalil Sima'an helped editing the article and provided guidance. Joachim Daiber wrote the article, developed all other software and performed all other experiments. Khalil Sima'an and Joachim Daiber produced the idea for the article.

All chapters of this thesis were written in full by the author.

Chapter 4

Examining the Relationship Between Preordering and Linguistic Typology

Preordering has seen a surge in popularity in statistical machine translation research in recent years, often providing reductions in translation time and showing good empirical results in translation quality. For many language pairs, however — especially for translation into morphologically rich languages — the assumptions of such models may be too crude. In this chapter, we study the relationship between typological aspects of a language pair, such as the word order freedom of the target language, and the effectiveness of preordering in statistical machine translation. We first provide an overview of current approaches to preordering and examine the linguistic motivations and limitations of the technique. We find that the assumptions of preordering can be insufficient for morphologically rich and free word order languages. While individual word order differences and morphological complexity are well-studied topics in linguistic theory, the notion of word order freedom is rarely addressed in a quantifiable way. To measure the word order freedom of languages in a quantitative manner, we therefore introduce a novel entropy measure which assesses how difficult it is to determine word order given a source sentence and its syntactic analysis. This measure, which we call bilingual head direction entropy, will enable us to examine the influence of word order freedom on the effectiveness of preordering in more detail in the following chapters.

Chapter Highlights

Problem Statement

- Preordering is a popular technique in statistical machine translation. While it has provided great benefits for some language pairs, it has been less successful for language pairs involving free word order and rich morphology.

Research Question

- Why does preordering not perform equally well across language pairs?
- Can we isolate and measure the typological factors that cause difficulties in some language pairs?

Research Contributions

- We examine the linguistic motivations and limitations of preordering and discuss how they relate to typological properties of the language pair.
- We focus on two important aspects of the target language, morphological complexity and word order freedom, and contribute a novel entropy measure that can quantify the word order freedom of the target language.

4.1 Introduction

A significant amount of research in machine translation has focused on methods for effectively restricting the often prohibitively large search space of statistical machine translation systems. One popular method providing a crude but theoretically motivated restriction of this space is preordering (also pre-reordering or source-side reordering). In preordering, the source sentence is rearranged to reflect the assumed word order in the target language. This provides an effective method for handling word and phrase movements caused by long-range dependencies, which usually enlarge the search space significantly. After preordering, decoding can be performed in fully monotone or close to monotone fashion, making the method applicable to a wide range of translation systems, including n-gram-based translation (Marino et al., 2006) and phrase-based machine translation. While systems using this approach have in the past not always been able to show improvements in translation quality over systems using more exhaustive search algorithms or specialized reordering models, preordering provides several benefits: Apart from facilitating the integration of additional information sources such as paraphrases, preordering approaches provide significant improvements in runtime performance. Jehl et al. (2014), for example, report an 80-fold speed improvement using

their preordering system compared to a standard system producing translations of the same quality.

The basic assumption inherent in the preordering approach is that it is feasible to predict target word order given only information from the source sentence. The majority of work on preordering uses a single permutation of the source sentence, which is passed on to the translation system. This leads to an even stronger assumption: it is feasible to predict a single preferred target word order. In this chapter we will discuss and evaluate whether this assumption is reasonable for all target languages. On the surface, this assumption seems reasonable for translating into fixed word order languages such as Japanese, but for translation into languages with less strict word order such as German, it is less likely to hold. In such languages there are often multiple plausible target word orders per source sentence because the underlying predicate-argument structure can be expressed with mechanisms other than word order (e.g. using morphological inflection or intonation). Hence, for these languages, it seems rather unlikely that choosing a single word order given only the source sentence can succeed. In this chapter, we want to examine the relationship between typological properties of the target language and the feasibility of preordering in more detail. Based on the findings of this chapter, Chapter 5 and 6 will then propose and evaluate potential solutions to dealing with these limitations.

We begin by examining the linguistic motivation and limitations of preordering (Section 4.2) and review common approaches to the task (Section 4.3). One of the major typological aspects influencing the effectiveness of preordering is the word order freedom of the target language. However, while individual differences in word order are well-studied, the notion of word order freedom is often difficult to define and quantify. We therefore contribute an information-theoretic measure to quantify the difficulty of predicting a target word order given the source sentence and its syntactic representation (Section 4.4). Our measure provides empirical support for the intuition that it is often not possible to predict a single word order for free word order languages, while it is more feasible for fixed word order languages such as Japanese.

4.2 Preordering: Linguistic Motivation and Limitations

If we consider translation as a generative process which transforms a source sentence $\mathbf{s}$ into a target sentence $\mathbf{t}$, we can separate this process into word order choices and lexical choices. Here, we denote word order choices as $\mathbf{t}_\pi$ and lexical choice as $\mathbf{t}_{\text{lex}}$ such that $\mathbf{t} = \langle \mathbf{t}_\pi, \mathbf{t}_{\text{lex}} \rangle$. If lexical choice and word order were fully independent, this could be conveniently expressed as $P(\mathbf{t} \mid \mathbf{s}) = P(\mathbf{t}_\pi \mid \mathbf{s}) P(\mathbf{t}_{\text{lex}} \mid \mathbf{s})$. However, this assumption is unrealistic and in practice preordering models fall back to a weaker assumption (4.2):

$$P(\mathbf{t} \mid \mathbf{s}) = P(\mathbf{t}_\pi, \mathbf{t}_{\text{lex}} \mid \mathbf{s}) \tag{4.1}$$

$$\stackrel{\text{def}}{=} P(\mathbf{t}_\pi \mid \mathbf{s}) P(\mathbf{t}_{\text{lex}} \mid \mathbf{t}_\pi, \mathbf{s}) \tag{4.2}$$

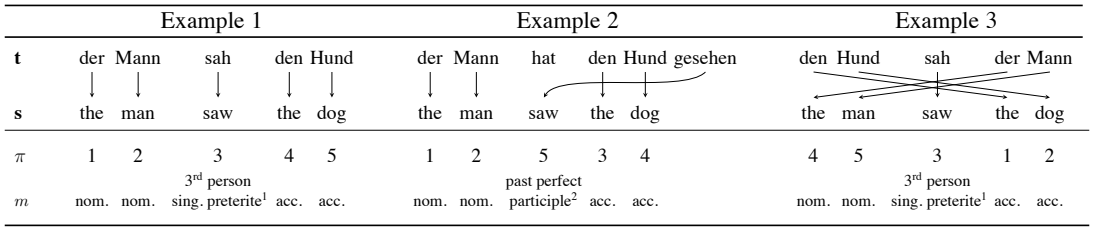


Figure 4.1: Example sentences from the language pair English–German.

The definition of $P(\mathbf{t}_\pi, \mathbf{t}_{\text{lex}} \mid \mathbf{s})$ is broken down into two parts: (a) word order prediction $P(\mathbf{t}_\pi \mid \mathbf{s})$ and (b) translation based on the predicted word order $P(\mathbf{t}_{\text{lex}} \mid \mathbf{t}_\pi, \mathbf{s})$. This process is commonly implemented in preordering by rearranging the source sentence $\mathbf{s}$ into target word order $\mathbf{s}'_\pi$:

$$P(\mathbf{t} \mid \mathbf{s}) = \sum_{\mathbf{s}'_\pi} P(\mathbf{s}'_\pi \mid \mathbf{s}) P(\mathbf{t} \mid \mathbf{s}'_\pi) \tag{4.3}$$

Preordering therefore abstracts away from the target side word order by combining $\mathbf{s}$ and $\mathbf{t}_\pi$ into the target-order source sentence $\mathbf{s}'_\pi$. In this model, lexical choice $\mathbf{t}_{\text{lex}}$ may depend on the word order choices but lexical choice cannot influence word order. This restriction is often addressed in an ad-hoc fashion by allowing the machine translation system to perform minimal reordering itself. To illustrate why this can be a problematic assumption, consider the English example sentence and its German translations in Figure 4.1. Between Examples 1 and 2, a change in the underlying grammatical structure (in this case the grammatical tense) causes a difference in both word choice and word order on the German side (*sah* vs. *hat gesehen*). This can be observed in the word order π and in the morphological attributes m in Figure 4.1: *sah* is a 3rd person singular preterite verb while *geschlagen* is a past participle verb. In this example, word order and word forms have to be selected conjointly in order to arrive at an adequate and fluent German translation.

Work in syntax-based machine translation and cross-lingual projection of syntactic annotation has demonstrated repeatedly that isomorphism between a source sentence and its translation on the syntactic level is limited (see, for example, the discussion of the *Direct Correspondence Assumption* in Hwa et al., 2005). On the other hand, this correspondence may be clearer on the level of each sentence's predicate-argument structure, and word order and morphology should be considered as two interchangeable means of realizing the underlying predicate-argument structure. This hypothesis is a central motivation for work in semantic transfer-based machine translation, such as the tectogrammatical approach discussed in Section 3.1. The success of preordering on morphologically impoverished target languages such as English therefore relies on the predicate-argument structure mostly being expressed via word order in these

[1]Full attributes in the data: `pos=verb, number=sing, person=3, verbform=fin, mood=ind, tense=past`.
[2]Full attributes in the data: `pos=verb, verbform=part, tense=past, aspect=perf`.

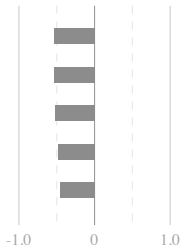
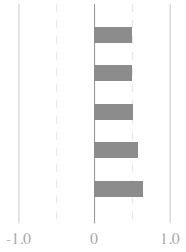
Correlation with preordering improvement		
WALS feature and value		Correlation
112A	Negative morphemes: Negative affix	
144A	Position of negative word: Morph. negation	
49A	Number of cases: 6-7 cases	
138A	Tea: Words derived from Sinitic cha	
15A	Weight-sensitive stress: Unbounded	
	...	
143E	Preverbal negative morphemes: NegV	
51A	Position of case affixes: No case affixes	
138A	Tea: Words derived from Min Nan Chinese te	
49A	Number of cases: no morph. case-marking	
50A	Asymmetrical case-marking: Add.-quant. asym.	

Table 4.1: Correlation (Pearson correlation coefficient) of preordering improvement (BLEU) with typological features of the target language.

languages. From a linguistic point of view, the effectiveness of preordering therefore seems fortuitous.

To illustrate this point further, consider Table 4.1, which shows the correlation between typological features of the target language and the improvements provided by preordering for a translation experiment between English and a diverse set of target languages (for more details on the experimental setup, see Chapter 9). Correlation is calculated for the relative difference between the BLEU score for each language with and without preordering using the Pearson correlation coefficient (Pearson, 1896).[3] The table shows that the language properties most correlated with successful preordering (WALS feature 49A, 50A, 51A) are highly indicative of a lack of rich morphology, while the properties most negatively correlated are properties commonly observed in morphologically rich languages, such as the use of affixes to express negation (WALS features 112A and 144A) or a significant number of cases (WALS feature 49A).

[3]Specifically, we calculate the correlation coefficient against the relative improvement of the BLEU score of the preordered system (BLEU_P) over the BLEU score of the baseline (BLEU_BL): $\frac{\text{BLEU}_\text{P} - \text{BLEU}_\text{BL}}{\text{BLEU}_\text{BL}}$. This ensures that the ranking is not influenced by the differences in the ranges of BLEU scores between languages, which due to the token-based formulation of BLEU are partially caused by morphological complexity themselves.

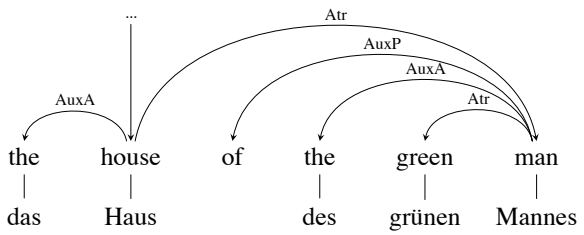


Figure 4.2: English dependency tree with aligned German translation.

4.3 Approaches to Preordering

Preordering has been explored from the perspective of the upper-bound achievable translation quality in several studies, including Khalilov and Sima'an (2012) and Herrmann et al. (2013), which compare various systems and provide oracle scores for syntax-based preordering models. Target-order source sentences, in which the word order is determined via automatic alignments, enable translation systems great jumps in translation quality and provide improvements in compactness and efficiency of downstream phrase-based translation models. Additionally, it was found that properties of the source syntax representation, such as how deeply phrase structure trees are nested, can significantly hamper the quality of these approaches. Approaches have largely followed two directions: (1) predicting word order based on some form of source-syntactic representation and (2) approaches which do not depend on source syntax.

Preordering with Source Syntax

Many approaches to preordering have made use of syntactic representations of the source sentence, including Collins et al. (2005) who restructure the source phrase structure parse tree by applying a sequence of transformation rules. More recently, Jehl et al. (2014) learn to order sibling nodes in the source-side dependency parse tree. The space of possible permutations is explored via depth-first branch-and-bound search (Balas and Toth, 1983). In later work, the authors further improve this model by replacing the logistic regression classifier with a feed-forward neural network (De Gispert et al., 2015), which results in improved empirical results and eliminates the need for feature engineering. Lerner and Petrov (2013) train classifiers to predict the permutations of up to six tree nodes in the source dependency tree. The authors found that by only predicting the best 20 permutations of n nodes, they could cover a large majority of the permutations in their data. Figure 4.2 shows an example dependency tree that can serve as input to such systems.

Preordering Without Source Syntax

Tromble and Eisner (2009) learn to predict the orientation of any two words (straight or inverted order) using a perceptron. The search for the best word order permutation is

performed with a $O(n^3)$ chart parsing algorithm. More basic approaches to syntax-less preordering include the application of multiple machine translation systems (Costa-jussà and Fonollosa, 2006), where a first system learns preordering and a second learns to translate the preordered sentence into the target sentence. Finally, there have been successful attempts at the automatic induction of parse trees from aligned data (DeNero and Uszkoreit, 2011) and the estimation of latent reordering grammars (Stanojević and Sima'an, 2015) based on permutation trees (Zhang and Gildea, 2007).

4.4 Quantifying Word Order Freedom

While specific differences in word order are a well-studied topic in linguistics and linguistic typology, word order freedom has only recently been studied from a quantitative perspective. This has been enabled partly by the increasing availability of syntactic treebanks. Kuboň and Lopatková (2015) propose a measure of word order freedom based on a set of six common word order types (SVO, SOV, etc.). Futrell et al. (2015) define various entropy measures based on the prediction of word order given unordered dependency trees. Both approaches require a dependency treebank for each language.

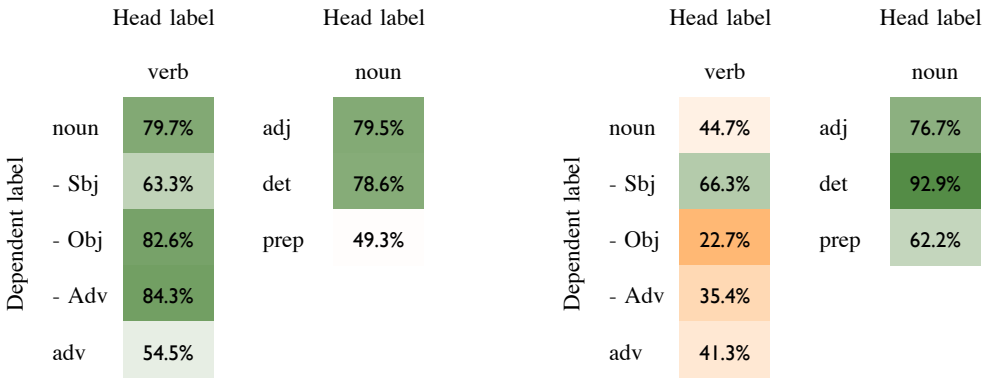
In practical applications such as machine translation, it is difficult to quantify the influence of word order freedom. For any arbitrary language pair, our goal is to quantify the notion of the target language's word order freedom based only on parallel sentences and source syntax. In their head direction entropy measure, Futrell et al. (2015) approach the problem of quantifying word order freedom by measuring the difficulty of recovering the correct linear order from a sentence's unordered dependency tree. We approach the problem of quantifying a target language's word order freedom by measuring the difficulty of predicting target word order based on the source sentence's dependency tree. Hence, we examine notions such as how difficult it is to predict French word order based on the syntax of the English source sentence.

4.4.1 Source Syntax and Target Word Order

We represent the target sentence's word order as a sequence of order decisions. Each order decision encodes for two source words, a and b, whether their translation equivalents are in the order (a, b) or (b, a). The source sentences are parsed with a dependency parser.[4] The target-language order of the words in the source dependency tree is then determined by comparing the target sentence positions of the words aligned to each source word. Figure 4.3 shows the percentage of dependent-head pairs in the source dependency tree whose target order can be correctly guessed by always choosing the more common decision.[5]

[4] `http://cs.cmu.edu/~ark/TurboParser/`

[5] For English–Japanese, we use manual word alignments of 1235 sentences from the *Kyoto Free Translation Task* (Neubig, 2011) and for English–German, we use a manually word-aligned subset of Europarl (Padó and Lapata, 2006) consisting of 987 sentences.



(a) English–Japanese (b) English–German

Figure 4.3: Source word pairs whose target order can be predicted fully using only the words' dependency relation or part-of-speech tags.

German and Japanese

Both language pairs differ significantly in how strictly the target language's word order is determined by the source language's syntax. English–German shows strict order constraints within phrases, such as that adjectives and determiners precede the noun they modify in the vast majority of cases (Figure 4.3b). However, English–German also shows more freedom on the clause level, where basic syntax-based predictions for the positions of nouns relative to the main verb are insufficient. For English–Japanese on the other hand, the position of the noun relative to the main verb is more rigid, which is demonstrated by the high scores in Figure 4.3a. These results are in line with the linguistic descriptions of both target languages. From a technical point of view, they highlight that any treatment of English–German word order must take into account information beyond the basic syntactic level and must allow for a given amount of word order freedom.

4.4.2 Bilingual Head Direction Entropy

While such a qualitative comparison provides insight into the order differences of selected language pairs, it is not straight-forward to compare across many language pairs. From the linguistic perspective, Futrell et al. (2015) use entropy to compare word order freedom in dependency corpora across various languages. They observed that artifacts of the data such as treebank annotation style can hamper comparability, but also found that a simple entropy measure for the prediction of word order based on the dependency structure can provide a reasonable quantitative measure of word order freedom.

We follow Futrell et al. (2015) in basing our measure on conditional entropy, which provides a straight-forward way to quantify to which extent target word order is deter-

mined by source syntax.

$$H(Y|X) = -\sum_{x \in X} P(x) \sum_{y \in Y} P(y|x) \log P(y|x) \tag{4.4}$$

Conditional entropy measures the amount of information required to describe the outcome of a random variable Y given the value of a second random variable X. Given a dependent-head pair in the source dependency tree, X consists of the dependent's and the head's part of speech, as well as the dependency relation between them. Note that as in all of our experiments the source language is English, the space of outcomes of X is the same across all language pairs. Y in this case is the word pair's target-side word order in the form of a decision between the order (a, b) or (b, a). Following Futrell et al. (2015), we estimate $H(Y|X)$ using the bootstrap estimator of DeDeo et al. (2013), which is less prone to sample bias than maximum likelihood estimation.[6]

Influence of Word Alignments

Futrell et al. (2015) use human-annotated dependency trees for each language they consider. Our estimation only involves word-aligned bilingual sentence pairs with a source dependency tree. Manual alignments are available for a limited number of language pairs and often only for a diminishingly small number of sentences. Consequently, the question arises whether automatic word alignments are sufficient for this task. To answer this question, we apply our measure to a set of manually aligned as well as a larger set of automatically aligned sentence pairs. In addition to the German and Japanese alignments mentioned above, we use manual alignments for English–Italian (Farajian et al., 2014), English–French (Och and Ney, 2003), English–Spanish (Graça et al., 2008) and English–Portuguese (Graça et al., 2008).

Since a limited number of manually aligned sentences are available, it is important to avoid bias due to sample size. Hence, we randomly sample the same number of dependency relations from each language pair. Considering only those languages for which we have both manual and automatic alignments, we can determine how well their word order freedom rankings correlate. Even though the number of samples for the manually aligned sentences is limited to 500 due to the size of the smallest set of manual alignments, we find a high Spearman's ρ correlation (Zwillinger and Kokoska, 1999) of $\rho = 0.77$ between the rankings of the six languages that occur in both sets.

Influence of Source Syntax

Another factor that may influence our estimated degree of word order freedom is the form and granularity of the source side's syntactic representation: More detailed representations may disambiguate cases that are difficult to predict with a more bare representation. As we are interested in the bilingual case and, specifically, in preordering, we

[6] We observe an average of 1033 values for X per language pair and perform 10000 Monte Carlo samples.

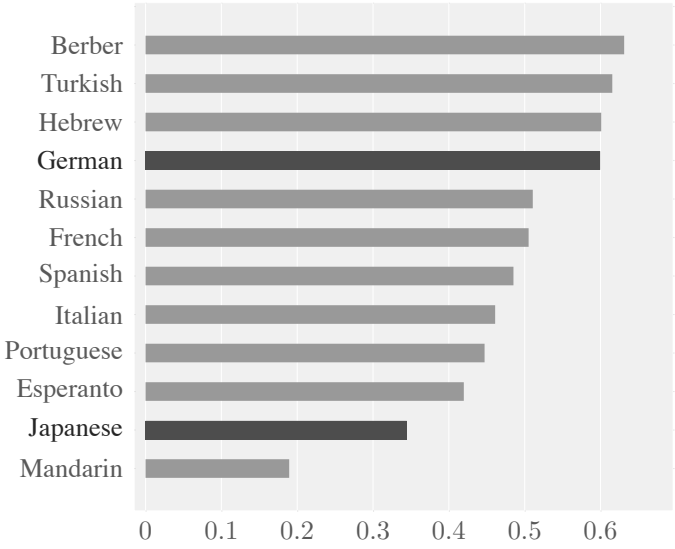


Figure 4.4: Bilingual head direction entropy with English source side.

content ourselves with using the same syntactic representation, i.e. dependency trees, that many preordering models use (e.g., Jehl et al. (2014), Lerner and Petrov (2013)).

Comparison to Monolingual Measures

Our measure is similar to Futrell et al. (2015)'s head direction entropy; however, it also offers several advantages. While monolingual head direction entropy requires a dependency treebank for each language, our bilingual head direction entropy only requires dependency annotation for the source language (English in our case). One of their caveats, the influence of the widely varying dependency annotation styles across treebanks, is also not present in our method since a single dependency style is used for the source language. We have demonstrated that automatic alignments perform on a comparable level to manual alignments. Accordingly, the amount of data that can be used to estimate the measure is only limited by the availability of parallel sentences. Finally, while dependency treebanks rarely cover the same corpora or even domains, our method can utilize sentences from the same or similar corpora for each language, thus minimizing potential corpus biases.

Translation from English

Figure 4.4 plots bilingual head direction entropy for an English source side and a set of typologically diverse target languages. For each language pair, we use 18000 sentence pairs and automatic alignments from the Tatoeba corpus (Tiedemann, 2012).[7]

[7]The alignments were produced using GIZA++ (Och and Ney, 2003) with *grow-diag-final-and* symmetrization.

Languages at the top of the plot in Figure 4.4 show a greater degree of word order freedom with respect to the English source syntax. Thus, predicting their word order from English source clues alone is likely to be difficult. In the next chapters, we will argue that in such cases it is crucial to pass on the ambiguity over the space of predictions to the translation model. By doing so, word order decisions can be influenced by translation decisions while still shaping the space of reachable translations.

4.5 Conclusion

In this chapter, we have examined the role of typological properties of the targeted language pair in the effectiveness of preordering in statistical machine translation. While preordering has shown significant benefits for some language pairs, it has not worked well for others. We have provided an overview over common approaches to preordering and discussed its linguistic motivations and limitations. We have observed that typological aspects of the target language, such as morphological complexity and word order freedom, can play a significant role in the effectiveness of preordering. For one particularly interesting aspect, word order freedom, we have provided a measure which enables an empirical comparison of language pairs in terms of the difficulty of predicting the target language's word order based on the source language. Our metric's predictions agree both with the intuition provided by linguistic theory and the empirical support in the form of translation experiments, which we will present in Chapter 6. While continuing to focus on the language pairs English–German and English–Japanese as language pairs representative of target languages with high and low degrees of word order freedom, we will propose and evaluate methods to address word order freedom in preordering in the following two chapters.

Chapter 5

Delimiting Morphosyntactic Search Space via Preordering Models: A Case Study

In the previous chapter, we have provided theoretical and empirical reasons why the effectiveness of preordering depends on typological properties of the languages it is applied to. Especially for morphologically rich and free word order target languages, the assumptions of these models seem ill-fitted. While such language pairs call for more complex models, these could in turn increase the search space to an extent that would diminish their benefits. One of the major issues of preordering is that in morphologically rich target languages, word order and word form choices often cannot be performed fully independently. In this chapter, we examine the question whether for such languages, models without any notion of morphology can be used as a means to delimit the search space for a machine translation system to a set of potential word order predictions instead of committing to just a single best order. We propose a novel preordering model based on a popular preordering algorithm (Lerner and Petrov, 2013), which is able to produce both n-best word order predictions as well as distributions over possible word order choices in the form of a lattice. We further show that the integration of non-local language model features can be beneficial for the model's preordering quality and evaluate the space of potential word order choices the model produces.

Chapter Highlights

Problem Statement

- The assumptions inherent in preordering models are ill-fitted for translation into morphologically rich and free word order target languages. For such languages, interaction between word order and morphology make selecting a single word order inadequate.

Research Question

- Can the limitations of preordering for translation into free word order languages be overcome by delimiting the space of potential word order choices instead of committing the machine translation system to a single best word order?

Research Contributions

- We propose a preordering model able to produce n-best word order predictions as well as distributions over possible word order choices in the form of a lattice.
- We show that the integration of non-local language model features is beneficial for the model's preordering quality.
- We show that using the space of potential word order choices the model produces is a promising approach for dealing with such language pairs.

5.1 Motivation

While for some languages preordering has provided great benefits, it has not performed equally well for other languages, including many morphologically rich languages. In this chapter, we evaluate a potential solution to this problem. We introduce a preordering model which can produce n-best word order predictions as well as distributions over possible word order choices and apply this model to the language pair English–German. The experiments with this language pair serve as a case study to examine the feasibility and effectiveness of using preordering models to delimit the space of potential word order choices so that the final word order decisions can be performed taking into account a broader range of signals. In the next chapter, we will consider more suitable ways to integrate such a model's predictions into the machine translation decoding process.

We begin by proposing a model and general framework for producing the space of potential word order choices in Section 5.2.1. In Section 5.2.2, we show how casting this model as context-free grammar parsing allows us to use cube pruning to integrate

non-local language model features. Our model is based on source syntax in the form of dependency trees. The reordering operations in source syntax-based approaches to preordering are often restricted by the form of the source-side syntactic trees; hence, the annotation conventions of the training treebank and the form of the predicted dependency trees play a significant role for the preordering system. We will therefore briefly describe the treebank format and other details of the experimental setup in Section 5.3.1. Section 5.3.2 and 5.3.3 present results of the experimental evaluation and a discussion of these results. We conclude in Section 5.4.

5.2 Delimiting Potential Word Order Choices

The goal of this chapter is to evaluate whether delimiting the space of potential word order choices provides a better alternative to committing to a single best word order. To examine this question, we first introduce a preordering model capable of producing n-best word order predictions and distributions over possible word order choices. Preordering systems can be compared along several dimensions. The main distinctions are whether the reordering rules are specified manually (Collins et al., 2005) or automatically learnt from data (Lerner and Petrov, 2013; Khalilov and Sima'an, 2012). Furthermore, approaches differ in the types of syntactic structures they assume. Systems may use either source or target syntax (Lerner and Petrov, 2013; Khalilov and Sima'an, 2012), both source and target syntax or no syntax at all (e.g. Stanojević and Sima'an (2015); DeNero and Uszkoreit (2011)). In this chapter, we focus on approaches using only source-side syntax. Dependency grammar offers a flexible and lightweight syntactic framework that can cover a large number of languages and provides suitable syntactic representations for reordering. Hence, we follow Lerner and Petrov (2013) in using dependency trees for the representation of source syntax.

5.2.1 Preordering Beyond First-Best Predictions

Our work is related to the work of Lerner and Petrov (2013), in which feature-rich discriminative classifiers are trained to directly predict the target-side word order based on source-side dependency trees. This is done by traversing the dependency tree in a top-down fashion and predicting the target order for each tree family (a family consists of a syntactic head and its children). To address sparsity issues, two models are introduced. For each subtree, the 1-step model directly predicts the target order of the child nodes. Unlike other preordering models, which often restrict the space of possible permutations, e.g. by the permutations permissible under the ITG constraint (Wu, 1997), the space of possible permutations for each subtree is restricted to the k permutations most commonly observed in the data. The blowup in permutation space with growing numbers of children is addressed by a second model, the 2-step model. This model decreases the number of nodes involved in any single word order decision. A binary classifier (called pivot classifier, in analogy to quicksort) first predicts whether a child

node should occur to the left or to the right of the head of the subtree. The order of the set of nodes to the left and to the right of the head is then directly predicted as in the 1-step model. In total, the 2-step approach requires one pivot classifier, 5 classifiers for the children on the left and 5 classifiers for the children on the right.

The cascade-of-classifiers approach used by this method (i.e. first predict the pivot, then predict the left and right orders, then recurse) exhibits the problematic characteristic that classification errors occurring near the top of the tree will propagate disproportionately to later decisions. The goal of the present work is to enable the preordering model to pass decisions on to a later stage. Hence, this error propagation issue may become problematic. In order to address this problem, we extract *n*-best word order predictions from each classifier. A distribution over the *n*-best preordered sentences can then be passed to a subsequent model or directly to a machine translation decoder either as a list of options or in the form of a lattice. Similar to the practice of *n*-best list extraction in machine translation decoders such as Moses, the preordering problem likewise allows the extraction of *n*-best preordering options either with or without additional integration of non-local features such as a language model.

General Model

We define a model over the possible orders of the tokens in the source sentence. Given a source sentence $\mathbf{s}$ and a corresponding dependency tree τ, π is the permutation of source tokens and π_h is a local permutation of a single tree family under head h. The score of a word order $\mathbf{s}'$ is:

$$P(\mathbf{s}' \mid \mathbf{s}, \tau) = \prod_{h \in \tau} P_T(\pi_h \mid \mathbf{s}, h, \tau), \tag{5.1}$$

where $P_T(\pi_h \mid \mathbf{s}, h, \tau)$ consists of decisions for the pivot and the left and right children:

$$P_T(\pi_h \mid \mathbf{s}, h, \tau) = P(\psi \mid \mathbf{s}, h, \tau) P_L(\pi_L \mid \mathbf{s}, h, \tau) P_R(\pi_R \mid \mathbf{s}, h, \tau) \tag{5.2}$$

For each dependency tree family, the generative story of this model is as follows: First, decide on whether each child node should go left or right of the head, i.e. $P(\psi \mid \mathbf{s}, h, \tau)$. Then, decide the order of the nodes to the left of the head, i.e. $P_L(\pi_L \mid \mathbf{s}, h, \tau)$, and to the right of the head, i.e. $P_R(\pi_R \mid \mathbf{s}, h, \tau)$.

Preordering Algorithm

Based on this model, we introduce the following preordering algorithm. For each source dependency tree family with head h, we extract the best k_T local word order predictions using the function PreorderFamily in Algorithm 1. $\Psi(cs)$ is the set of possible choices when distributing nodes using the pivot classifier. Given a set of child nodes cs, $\Pi(cs)$ is the set of their possible permutations. The best permutations for the left and right side are extracted by the following methods:

$$\hat{\pi}_L \leftarrow \underset{\pi_L \in \Pi(cs_L)}{\arg\,\text{bestk}}\ P_L(\pi_L \mid \mathbf{s}, h, \tau) \quad (5.3) \qquad \hat{\pi}_R \leftarrow \underset{\pi_R \in \Pi(cs_R)}{\arg\,\text{bestk}}\ P_R(\pi_R \mid \mathbf{s}, h, \tau) \quad (5.4)$$

Since this model is implemented using multi-class classifiers, finding the best k_O permutations for the nodes to the left and right of the head, i.e. Equation 5.3 and 5.4, only require one multi-class classification. Following Lerner and Petrov (2013), we restrict the set of allowed permutations $\Pi(cs)$ to the 20 most common permutations observed in the training data. Given a pivot decision $\hat{\psi}$ (which children go left and which go right of the head?), $\textsc{Left}(\hat{\psi})$ returns the children to the left and $\textsc{Right}(\hat{\psi})$ returns the children to the right of the head. The function $\textsc{Permutation}(\hat{\psi}, \hat{\pi}_L, \hat{\pi}_R)$ returns the word order permutation resulting from the pivot decision, the left children order and the right children order.

Algorithm 1 n-best preordering of a source tree family.

procedure PREORDERFAMILY(h, τ)
  $cs \leftarrow$ CHILDREN(h, τ)
  $topk \leftarrow$ PRIORITYQUEUE()

  for $\hat{\psi} \leftarrow \underset{\psi \in \Psi(cs)}{\arg\,\text{bestk}}\ P(\psi \mid \mathbf{s}, h, \tau)$ **do** ▷ Pivot decisions
    $cs_L \leftarrow \textsc{Left}(\hat{\psi})$
    $cs_R \leftarrow \textsc{Right}(\hat{\psi})$
    for $\hat{\pi}_L \leftarrow \underset{\pi_L \in \Pi(cs_L)}{\arg\,\text{bestk}}\ P_L(\pi_L \mid \mathbf{s}, h, \tau)$ **do** ▷ Left order decisions
      for $\hat{\pi}_R \leftarrow \underset{\pi_R \in \Pi(cs_R)}{\arg\,\text{bestk}}\ P_R(\pi_R \mid \mathbf{s}, h, \tau)$ **do** ▷ Right order decisions
        $p \leftarrow \textsc{Permutation}(\hat{\psi}, \hat{\pi}_L, \hat{\pi}_R)$
        $topk.\textsc{push}(P(\hat{\psi} \mid \mathbf{s}, h, \tau) \times P_L(\hat{\pi}_L \mid \mathbf{s}, h, \tau) \times P_R(\hat{\pi}_R \mid \mathbf{s}, h, \tau), p)$
  return $topk.\textsc{take}(k_T)$

For n children, there are $S(n, 2)$ possible pivot decisions, where $S(n, k)$ is the Stirling number of the second kind. Since this number grows exponentially with n, it would be extremely expensive, if not infeasible, to consider all possible pivot decisions. Hence, similar to the extraction of $\hat{\pi}_L$ and $\hat{\pi}_R$, the extraction of the possible choices for the pivot decision, i.e. $\hat{\psi}$, is implemented as k-best Viterbi extraction from a conditional random field classifier: $\hat{\psi} \leftarrow \underset{\psi \in \Psi(cs)}{\arg\,\text{bestk}}\ P(\psi \mid \mathbf{s}, h, \tau)$.

This approximation means that only the best k_P pivot decisions are considered. Hence, for each of the maximally k_P possible ways to distribute the child nodes when taking the pivot decision, two classifications have to be performed: one for the nodes on the left and one for the nodes on the right. The extraction of *n*-best word order predictions therefore requires $2 \times k_P$ classifications for each source-side tree family. With the best k_T local permutations for each source tree family, we can then extract *n*-best permutations for the whole tree. If all order decisions in this model are local

to their tree family, extracting the best permutations for the whole sentence is straightforward. In the next section, we will discuss how this assumption changes with the introduction of non-local features.

5.2.2 Integration of Non-Local Features

While the basic model introduced by Lerner and Petrov (2013) shows promising empirical performance, it also makes fairly strong independence assumptions. The generative process assumes that the local order decisions occur only within individual tree families defined by the dependency tree. Hence, a local word order decision at any point in the dependency tree is fully independent from any other decision in the tree. For languages such as German, this independence assumption can be problematic because the position of a constituent in the sentence influences the internal word order.[1] For example, certain positions allow for scrambling, i.e. more or less free movement of some constituents within a specific area of the sentence. Previous work on preordering (Khalilov and Sima'an, 2012) has shown that the integration of even a weak trigram language model estimated over the gold word order predictions $\mathbf{s}'$ can improve preordering performance. Since we use projective dependency trees, which are internally converted to a flat phrase structure representation, the model can be expressed in the form of a weighted context-free grammar in which labels encode the order of the constituents. One method to weaken the independence assumptions of this grammar is the direct integration of a language model (LM). This idea is reminiscent of the integration of the finite state language model with the synchronous context-free grammar used in hierarchical phrase-based machine translation (Chiang, 2007).

Hence, instead of searching for $\hat{\mathbf{s}}' = \arg\max_{\mathbf{s}'} P(\mathbf{s}' \mid \mathbf{s}, \tau)$, the search will now include the n-gram language model, such that: $\hat{\mathbf{s}}' = \arg\max_{\mathbf{s}'} P(\mathbf{s}' \mid \mathbf{s}, \tau) P_{LM}(\mathbf{s}')$. This integration can be performed in three ways: the simplest form of integration, which is fast but allows for significant search errors, is to generate an *n*-best list of word order predictions using the $-$LM preordering model (i.e., without the LM or other non-local features) and re-score this list using the language model. On the other end of the spectrum, the language model can be integrated by performing a full intersection between the preordering CFG and the finite state automaton that defines the language model (Bar-Hillel et al., 1961). While this would allow for exact search, this method is found to be too slow in practice. A compromise between these two extremes is cube pruning (Chiang, 2007), in which the inner LM cost as well as the left and right LM states are stored on each node, so that it is possible to perform bottom-up dynamic programming to efficiently determine the total LM cost by combining the intermediate node costs. Keeping the properties required for performing cube pruning, we use the more general log-linear model formulation (Och and Ney, 2002) by defining the search for the best

[1] German word order is generally described based on a set of topological fields in which constituents are placed and which restrict their movement (see Müller, 2015).

word order prediction $\hat{\mathbf{s}}'$ as follows:

$$\hat{\mathbf{s}}' = \arg\max_{\mathbf{s}'} P(\mathbf{s}' \mid \mathbf{s}, \tau)^{\lambda_{RM}} P_{LM}(\mathbf{s}')^{\lambda_{LM}} \tag{5.5}$$

$$= \arg\max_{\mathbf{s}'} \prod_i \phi_i(\mathbf{s}', \mathbf{s}, \tau)^{\lambda_i} \tag{5.6}$$

$$= \arg\max_{\mathbf{s}'} \sum_i \lambda_i \log \phi_i(\mathbf{s}', \mathbf{s}, \tau), \tag{5.7}$$

where $\phi_i(\mathbf{s}', \mathbf{s}, \tau)$ is the ith feature function and λ_i is its weight. On every source tree node, cube pruning is performed with a beam size of $k_{+\text{LM}}$. The best $k_{-\text{LM}}$ preordering labels are considered for expansion. Additionally, we prune all preordering labels for which the language model cost is higher than the language model cost of the original source tree order (i.e., performing no reordering). To make individual configurations comparable, we follow Chiang (2007) in adding a heuristic cost that approximates the cost of the first $m-1$ words: $\log P_{\text{LM}}(\mathbf{s}'_1 ... \mathbf{s}'_l)$ where $l = \min\{m-1, |\mathbf{s}'|\}$ for an m-gram language model. In our case, $\mathbf{s}'$ is the vector of preordered source-side words at a specific tree node. We add the heuristic cost of all relevant feature functions ϕ_i for the set of language model feature functions Φ_{LM} as $\sum_{i \in \Phi_{\text{LM}}} \lambda_i \log \phi_i(\mathbf{s}'_1 ... \mathbf{s}'_l)$.

Feature Functions

The log-linear model formulation makes the addition of arbitrary local and non-local features possible; hence, any suitable feature function can be added to this model. We use the following initial features.

Lexicalized preordering model The most important feature is the lexicalized preordering model $P(\mathbf{s}' \mid \mathbf{s}, \tau)$ introduced in Section 5.2.1. We call this model lexicalized since it makes decisions based on the source words while other models might make predictions based on non-lexical information (e.g., part-of-speech tags).

Language models To weaken the strong independence assumptions of this model, we add a generic n-gram language model over the gold word order predictions $\mathbf{s}'$, a language model over part-of-speech tags and a class-based language model.

Unlexicalized preordering model As the lexicalized preordering model might run into sparsity issues, we add as a further feature function a weaker model $P_W(\pi \mid h, cs)$, where cs is the set of children represented by their dependency label and by whether they have children, and h is the head represented by its POS tag. The model is estimated via maximum likelihood estimation from the oracle word order choices restricted by the source-side dependency trees (*oracle tree reordering*). These tree-restricted oracle word order choices differ from the free oracle word order choices in that words are not allowed to move out of the constituents of the dependency tree. For example, in the

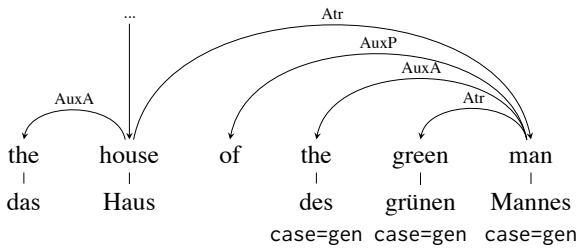


Figure 5.1: Syntactic representation of English PP as German genitive NP translation.

English sentence "the house of the green man" in Figure 5.1, the word "green" would always be on the same side of "house" as "man" since as a dependent of "man", it will always move with "man" in relation to its grandparent "house."

5.2.3 Applicability of the Model

While we focused on one particular *n*-best preordering method in Section 5.2.1, the general model introduced in Section 5.2.2 is applicable to any preordering model over source trees for which *n*-best candidates can be extracted. For example, the pairwise neural network-based method by De Gispert et al. (2015) can be used either by extracting *n*-best decisions directly from the graph or by applying the CKY algorithm on the space of permutations permissible under ITG (Tromble and Eisner, 2009).

5.3 Experiments

In this section, we present experimental results for the language pair English–German. First, we will describe the details of the preordering system and the experimental setup and highlight assumptions and decisions made in the system. After introducing the experimental setup, we will turn to examine the following two questions. In Section 5.2.2, we have shown how cube pruning can be used to integrate non-local features into the preordering model. We will therefore first consider the question whether the preordering model benefits from the access to non-local features that this provides. Subsequently, we will turn to evaluating the quality of the space of potential word order choices produced by the model and discuss whether using the preordering model to delimit this space can provide a better alternative to committing to a single word order prediction for morphologically rich target languages.

5.3.1 Implementation and Experimental Setup

Source-Side Syntax

For source syntax-based preordering to work reliably, the dependency representation should fulfill certain requirements: Flatter trees increase the space of coverable per-

mutations while the information in the neglected segmentations may be recoverable by the preordering model. Additionally, whenever reasonable, content-bearing elements should be treated as the head.[2] We use a customized version of the treebank collection and transformation tool HamleDT (Zeman et al., 2012) for this purpose.

Model Training

For training the model, we mostly follow the process from Lerner and Petrov (2013). Training instances are extracted from the automatically aligned training data based on a small set of manually defined rules. To ensure high-quality training data, only subtrees that are fully connected by high-confidence alignments are considered. The preordering classifiers are trained on the intersection of high-confidence word alignments and the first-best output of the TurboParser dependency parser (Martins et al., 2009). The alignments are created using the Berkeley aligner[3] with the hard intersection setting. While using only high-confidence alignment links will lead to a reduction in the number of alignment links, it creates more reliable training data for the preordering model. The dependency parser is trained to produce pseudo-projective dependency trees (Nivre and Nilsson, 2005).[4] Appropriate values for $k_{+\text{LM}}$ and $k_{-\text{LM}}$ are determined using grid search. We found that beam sizes above $k_{+\text{LM}} = 15$ and $k_{-\text{LM}} = 5$ did not improve first-best preordering quality.

Model Tuning

The set of weights λ for the combination of the preordering model and the language models are optimized for a selected target metric on holdout data. The straight-forward choice for this metric is Kendall τ, which indicates the similarity of the word order of both sides. The Kendall τ distance $d_\tau(\pi, \sigma)$ between two permutations π and σ is defined as (Birch et al., 2010):

$$d_\tau(\pi, \sigma) = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij}}{Z}, \tag{5.8}$$

$$\text{where } z_{ij} = \begin{cases} 1 & \text{if } \pi(i) < \pi(j) \text{ and } \sigma(i) > \sigma(j) \\ 0 & \text{otherwise} \end{cases} \quad \text{and } Z = \frac{(n^2 - n)}{2}. \tag{5.9}$$

The metric indicates the ratio of pairwise order differences between two permutations. An alternative to this ordering measure would be the simulation of a full machine translation system, as first proposed by Tromble and Eisner (2009). To ensure that the changes in word order do not affect this mock translation system and to limit its complexity, such a system would be limited to phrases of length 1.

[2] For example, auxiliary verbs should modify the finite verb and prepositions should depend on the head of the noun phrase.

[3] `https://code.google.com/p/berkeleyaligner/`

[4] Projectivization was performed using MaltParser version 1.8; `http://www.maltparser.org/`.

Model	Kendall τ	BLEU ($\hat{\mathbf{s}}' \rightarrow \mathbf{s}'$)
First-best $-$LM	92.16	68.1
First-best $+$LM (cube pruned)	92.27	68.7
Best out of n-best $+$LM (cube pruned, $n = 5$)	93.33	–
Best out of n-best $+$LM (cube pruned, $n = 10$)	93.72	–

Table 5.1: LM integration tested on first-best prediction. English–German, scores from predicted English ($\hat{\mathbf{s}}'$) to gold-ordered English ($\mathbf{s}'$).

We perform tuning towards Kendall τ using the *tuning as ranking* (PRO) framework (Hopkins and May, 2011). At tuning time, $k_{-\text{LM}}$ and $k_{+\text{LM}}$ are set to 15 and 100 respectively. PRO requires the unweighted values of all feature functions; hence, during tuning only, we retain the unweighted feature values on each node and sum over intermediate values to arrive at the overall scores. Training instances for ranking are sampled from the best 100 word order predictions for each sentence in the tuning set. We perform 6 iterations and interpolate the weights of each iteration with the weights from the previous iteration by the recommended factor of $\Psi = 0.1$.

Translation Setup

To evaluate the model in a full translation setup, we follow the standard approach to preordering. Given the source side **s** and the target side **t** of the parallel training corpus, we first perform word alignment using MGIZA++ (Gao and Vogel, 2008). We perform 6 iterations of IBM Model 1 training followed by 6 iterations of HMM word alignment and 3 iterations each of IBM Model 3 and 4.

After initial training, the preordering model is applied to **s**, obtaining the preordered corpus $\hat{\mathbf{s}}'$. Since the word order differences between $\hat{\mathbf{s}}'$ and **t** should be less acute, less computationally expensive word alignment tools are sufficient to re-align the corpus. We align $\hat{\mathbf{s}}'$ and **t** using `fast_align`,[5] an efficient re-parameterization of IBM Model 2 (Dyer et al., 2013). Improvements in word order can lead to improvements in alignments and hence the training and word alignment process can be performed repeatedly. Lerner and Petrov (2013) report no significant improvements after the initial re-alignment. Accordingly, we do not iterate the training process either. The underlying translation system is Moses (Koehn et al., 2007) using the standard feature setup and using only the distortion-based reordering model (with a distortion limit of 7). Tuning is performed using MERT (Och, 2003). The system is trained on the full parallel sections of the Europarl corpus (Koehn, 2005) and tuned and tested on the WMT 2009 and WMT 2010 newstest sets respectively. The language model is a 5-gram n-gram model trained on the target side of Europarl and the news commentary corpus.[6]

[5] `http://github.com/clab/fast_align`

[6] `http://statmt.org/wmt13/translation-task.html`

	Distortion limit	BLEU	METEOR	TER
Baseline	7	15.20	35.43	66.62
Best out of k ($k = 10$)		17.26[A]	37.97[A]	62.64[A]

[A] Result is statistically significant against baseline at $p < 0.05$.

Table 5.2: Estimation of the quality of the k best word order predictions.

5.3.2 Testing the Effectiveness of Non-Local Features

While our preliminary results showed that the integration of a language model might be helpful, we now consider this question in more detail. To test whether the language model features are beneficial to the reordering model, we compare two versions of the same system: *first-best* −LM is the reordering system without a language model and *first-best* +LM is the same system with the language model integrated via cube pruning. Results are presented in Table 5.1. While Kendall τ gives an impression of the overall word order quality, the BLEU metric applied to the reordered source sentences gives an indication of the quality of reordering within the more restricted space of the length of the n-grams used in the metric. The results show that the integration of the language model helps the system improve the quality of the word order predictions. We expected the language model to provide benefits mostly on the borders between tree nodes. The BLEU score indicates an improvement in the ordering of short word sequences, which hints at the presence of this benefit.

In the "Best out of n-best +LM" setup in Table 5.1, we produce the top n word order predictions and select the prediction that provides the most Kendall τ improvement. These results hint at the potential improvement contained in the best n predictions of the model. Next, we turn to examining the quality of the space of word order predictions in more detail by applying them in a machine translation task.

5.3.3 Evaluating the Quality of the Word Order Predictions

Our goal in this chapter has been to use a preordering model to delimit the search space for a subsequent, more complex model. Hence, in order to examine the model presented in Section 5.2, we determine the quality of the n-best predictions the model produces. We perform the following experiment for the language pair English–German: Using the preordering system, we produce the 10 best word order predictions for each sentence in the test set. We then translate each sentence arranged according to each of the word order predictions using a standard phrase-based machine translation system trained on the corpus produced by the first-best preordering system. After the translation is performed, a single translation is selected based on the best sentence-level BLEU score. Table 5.2 shows results for this setup and for a baseline system without preordering. Both systems use a distortion limit of 7 and use only the standard distance-based reordering model. Statistical significance tests are performed using bootstrap resampling (Koehn, 2004a) and statistically significant results ($p < 0.05$) are marked with the let-

ter A. The results show that significant improvements in translation quality measured in terms of BLEU, METEOR and TER are possible based on the space of word order choices provided by our model.

5.3.4 Discussion

Having introduced our preordering method and having evaluated the influence of non-local features, we are interested in two basic aspects of the output space provided by this system: The first aspect is the quality of the space delimited by the preordering system. Since we plan to pass the output space to a subsequent translation model, it has to be ensured that a sufficient number of good candidates are contained in this space. This question is answered by the translation experiments performed in Section 5.3.3, which indicate that even within the first 10 word order predictions per sentence, there are enough good instances to enable a significant improvement in translation quality. Since the evaluation of our translation experiments is performed using only automatic evaluation metrics, it is difficult to pinpoint the exact source of these potential improvements. In the next chapter, we further examine this aspect by performing broader experiments with a preordering space in the form of a lattice. The second question is whether the size of the space of potential word order choices is manageable for subsequent models. Since our experiments show that even with only 10 word order predictions, a significant improvement can be observed, it is clear that this very small space can be used by a subsequent model. In addition to this, the output in the form of a lattice allows for using a larger number of options and efficient processing using dynamic programming algorithms.

5.4 Conclusion

Preordering provides significant potential for improvements in translation quality and translation performance in machine translation, which was shown in previous studies and is supported by the method's recent surge in popularity. Most of the benefits of preordering are due to enabling the modeling of much larger reordering spaces in a more reliable manner than it would be possible within the underlying machine translation system. For target languages such as German or Arabic, however, word order and morphology are interconnected and should not be treated in isolation. As a first step towards broader morphosyntactic processing beyond word order only, this chapter has explored how a preordering model can be utilized to produce a space of sensible word order predictions. We have presented a novel preordering model for this purpose and have evaluated its outputs with translation experiments using a common system setup. Our experiments show that non-local language model features integrated via cube pruning improve the preordering quality for the language pair English–German. Further, our translation experiments show that this preordering system, when optimized for producing n-best predictions, provides an output space that is valuable for further processing

both in its compactness and in the potential improvement in translation quality it enables. This chapter has served as an exploration into the merit of using preordering models to delimit the space of potential word order choices for machine translation. In the next chapter, we will build on this idea by using preordering models to generate lattices of potential word order predictions. Such lattices can be integrated directly into the decoding process of a phrase-based machine translation system, thus enabling the use of a broader range of non-local signals in the decoder.

Chapter 6

Machine Translation with Word Order Permutation Lattices

We address preordering for two target languages at the far ends of the word order freedom spectrum, German and Japanese. For languages with more word order freedom, attempting to predict a single word order given only the source sentence seems less suitable; therefore, we examine solutions which fit both strict word order and free word order target languages. In Chapter 5, we observed that delimiting the space of word order choices provides a potential solution for free word order target languages and that non-local features can support the preordering model in making good word order choices. A more general approach to this initial exploration would be to pass the uncertainty of the preordering model on to the machine translation decoder, which can then perform decisions while taking into account a broader set of signals. Thus, we examine lattices of n-best word order predictions as a unified representation for typologically diverse target languages. We present an effective solution to the resulting technical issue of how to select a suitable source word order from the lattice during training. Our experiments show that lattices are crucial for good empirical performance for languages with freer word order (English–German) and can provide additional improvements for fixed word order languages (English–Japanese).

Chapter Highlights

Problem Statement

- A suitable representation of word order choices should provide benefits for both strict and free word order target languages. However, passing a single word order prediction to the machine translation system is insufficient for target languages with greater word order freedom.

Research Question

- Can word order permutation lattices provide a suitable representation for word order choices for typologically diverse target languages?

Research Contributions

- We propose lattices of n-best word order predictions by preordering models as a unified representation of word order choices.
- We present an effective solution to the technical issue of how to select an appropriate source word order from the lattice during training.
- We perform experiments for English–German and English–Japanese, establishing word order permutation lattices as a suitable representation for typologically diverse target languages.

6.1 Motivation

Word order differences between a source and a target language are a major challenge for machine translation systems. For phrase-based models, the number of possible phrase permutations is so large that reordering must be constrained locally to make the search space for the best hypothesis tractable. However, constraining the space locally incurs the risk that the optimal hypothesis becomes unreachable. Preordering of the source sentence has been embraced as a way to ensure the reachability of certain target word order constellations for improved prediction of the target word order. Preordering aims at predicting a permutation of the source sentence which has minimal word order differences with the target sentence; the permuted source sentence is then passed on to a translation system trained to translate target-order source sentences into target sentences. In the previous two chapters, we have examined whether a basic assumption inherent in the preordering approach, namely that it is feasible to predict target word order given only information from the source sentence, is reasonable for all languages. We concluded that while the assumption seems reasonable for translating into fixed

word order languages such as Japanese, for languages with less strict word order such as German, it is less likely to hold. In this chapter, we want to examine the relationship between a target language's word order freedom and preordering in more detail.

Based on the idea of delimiting the space of potential word order choices, which we examined for the language pair English–German in the previous chapter, we study the option of passing n-best word order predictions, instead of a single word order, to the translation system as a lattice of possible target word order choices for the source sentence. For the training of the translation system, the use of such permutation lattices raises a question: How should the training corpus for a lattice-preordered translation system be prepared? In previous work on standard preordering using single word order predictions, the training data consists of pairs of source and target sentences where the source sentence is either in target order (i.e. ordered based on word alignments) or preordered (i.e. in predicted order). In this chapter we contribute a novel approach for selecting training instances from the lattice of word order permutations: We select the permutation in the lattice providing the best match with the target-order source sentence (we call this process "lattice silver training").

Our experiments show that for English–Japanese and English–German lattice preordering has a positive impact on the translation quality. While lattices enable further improvement for preordering English into the strict word order language Japanese, lattices in conjunction with our proposed lattice silver training scheme turn out to be crucial to reach satisfactory empirical performance for English–German. This result highlights that when predicting the word order of free word order languages given source-side information only, it is important to ensure that the word order predictions and the translation system can interact sufficiently.

6.2 Lattice Translation

In this section, we introduce related work on lattice-based machine translation. For an overview of preordering, see Section 4.3 of Chapter 4.

A lattice is a directed acyclic graph with a single starting point and can be interpreted as an acyclic finite-state automaton defining a finite language. A special case of lattices, confusion networks (also called sausage, Bertoldi et al., 2007), have been extensively used for representing alternative input sequences in various natural language processing tasks. In a confusion network, every path from the start node to the final node passes through all other nodes. Applications have mostly focused on representing intermediate hypotheses in tasks such as speech translation (Ney, 1999; Bertoldi et al., 2007) or parsing of noisy input text (van der Goot and van Noord, 2017), or to account for ambiguity due to pre-processing (Xu et al., 2005; Dyer, 2007). In machine translation, lattices have been used to delimit the space of permutations of the input considered by the decoder in a few instances (Knight and Al-Onaizan, 1998; Kumar and Byrne, 2003). Word order permutation lattices have been demonstrated to be effective by Zhang et al. (2007). However, except for n-gram based decoders (Khalilov

et al., 2009) this approach is not common practice.

Lattice translation for phrase-based and hierarchical phrase-based machine translation was first formalized by Dyer et al. (2008). Phrase-based models require a modification of the standard decoding algorithm to maintain a coverage vector over states, rather than input word positions. In standard phrase-based decoding with a distortion limit, the complexity of the space of translation options is $O(\text{max stack size} \times \text{sentence length})$ and the number of translation options considered by the decoder is reduced using beam search (see Sections 2.3.4 and 2.3.5 of this thesis). With lattice input, the complexity depends on $|Q|$, where Q is the set of states of the lattice, instead of the input sentence length. Due to stack decoding, the number of translation options explored by the decoder is independent of the number of transitions in the lattice. As in standard phrase-based decoding, the states of a lattice can be visited non-monotonically. Dyer et al. (2008) propose to estimate the distance between two nodes as the length of the shortest path between them. The shortest path can be pre-calculated using an all-pairs shortest path algorithm prior to decoding (see e.g. Chapter 25 of Cormen et al., 2001). The Moses machine translation decoder used in this chapter uses the $O(|Q|^3)$ Floyd-Warshall algorithm for this purpose. As in standard decoding, a *distortion limit* is used to ensure that the translation space remains tractable.

In this chapter, we use lattice input to constrain the space of permutations of the source sentence allowed within the decoder. Additionally, we completely disable the decoder's subsequent reordering capabilities in most cases. Because our models can perform global permutation operations without ad-hoc distortion limits, we can reach far more complex word orders. Crucially, our models are better predictors of word order than standard distortion-based reordering, thus we manage to decode with relatively small permutation lattices.

6.3 Preordering Free and Fixed Word Order Languages

The measure of word order freedom introduced in Chapter 4 enables us to estimate how difficult it is to predict the target language's word order based on the source language. In this section, we introduce the two preordering models we use to predict the word order of German and Japanese. Experiments with these models will allow us to examine the relationship between preordering and word order freedom in an empirical setting and enable us to test the suitability of word order permutation lattices as a typologically robust representation of word order choices.

6.3.1 Neural Lattice Preordering

Based on their earlier work, which used logistic regression and graph search for preordering (Jehl et al., 2014), De Gispert et al. (2015) introduce a neural preordering model. In this model, a feed-forward neural network is trained to estimate the swap probabilities of nodes in the source-side dependency tree. Search is performed via the

depth-first branch-and-bound algorithm. The authors have found this model to be fast and to produce high-quality word order predictions for a variety of languages.

Model Estimation

Training examples are extracted from all possible pairs of children of a dependency tree node, including the head itself. For each pair, the two nodes are swapped if swapping them reduces the number of crossing alignment links. The crossing score of two nodes a and b (a precedes b in linear order) and their aligned target indexes A_a and A_b is defined as follows:

$$\text{cs}(a, b) = |\{(i, j) \in A_a \times A_b : i > j\}|$$

Training instances generated in this manner are then used to estimate the swap probability $p(i, j)$ for two indexes i and j. For each node in the source dependency tree, the best possible permutation of its children (including the head) is determined via graph search. The score of a permutation of length k is defined as follows:

$$\text{score}(\pi) = \prod_{1 \leq i < j \leq k | \pi[i] > \pi[j]} p(i, j) \prod_{1 \leq i < j \leq k | \pi[i] < \pi[j]} 1 - p(i, j) \tag{6.1}$$

We closely follow De Gispert et al. (2015) for the implementation of the estimator of $p(i, j)$. A feed-forward neural network (Bengio et al., 2003) is trained to predict the orientation of a and b based on a sequence of 20 features, such as the words, the words' POS tags, the dependency labels, etc.[1] The network consists of 50 nodes on the input layer, 2 on the output layer, and 50 and 100 on the two hidden layers. We use a learning rate of 0.01, a batch size of 1000 and perform 20 training epochs.

Search

Search in this model consists of finding the sequence of swaps leading to the best overall score according to the model. Let a partial permutation of k nodes be a sequence of length $k' < k$ containing each integer in $\{1, ..., k\}$ at most once. The score of a new permutation obtained by extending a partial permutation π' of length k' by one element can be computed efficiently as:

$$\begin{aligned} \text{score}(\pi' \cdot \langle i \rangle) = \ & \text{score}(\pi') \\ & \prod_{j \in V | i > j} p(i, j) \\ & \prod_{j \in V | i < j} 1 - p(i, j) \end{aligned} \tag{6.2}$$

Algorithm 2 k-best branch-and-bound search.

Precondition: m: maximum sequence length, ϵ: empty sequence, $bound_0$: initial bound, k: maximum number of permutations

```
procedure KBestBNB(ε, bound_0, m)
    bestk ← ∅
    bound ← bound_0
    Search(⟨ε⟩)
    return bestk

procedure Search(π')
    if score(π') > bound then
        if |π'| = m then
            if |bestk| = k then
                remove worst permutation in bestk
            bestk ← bestk ∪ ⟨π'⟩
            bound ← score(π')
            if |bestk| = k then
                bound ← worst permutation in bestk
            return
        else
            for each i ∈ {1, ..., m} \ π' do
                Search(π' · ⟨i⟩)
```

k-Best Search

Target languages such as German allow for a significant amount of word order freedom; hence, the depth-first branch-and-bound algorithm, which extracts the single best permutation, may not be the best choice in this case. In the context of the Traveling Salesman Problem, van der Poort et al. (1999) show that general branch-and-bound search can be extended to retrieve k-best results while keeping the same guarantees and computational complexity. Only minor changes are necessary to adapt the search for the best permutation to finding the k-best permutations: We keep a set $bestk$ of the best permutations and a single $bound$. If for a permutation π', score$(\pi') > bound$, instead of updating the bound to the single best permutation and remembering it, the following steps are performed:

1. If $|bestk| = k$:
   - Remove worst permutation from the set.
2. Add π' to $bestk$.

[1] Our implementation is based on `http://nlg.isi.edu/software/nplm/`.

3. The new $bound$ will be the score of the worst permutation in $bestk$.

Pseudocode for the full algorithm is presented in Algorithm 2.

6.3.2 Reordering Grammar Induction

Reordering Grammar is a hierarchical unsupervised approach to preordering proposed by Stanojević and Sima'an (2015). In this approach, a probabilistic context-free grammar is induced from aligned parallel data. The resulting grammar can predict permutation trees (Zhang and Gildea, 2007), which are a form of constituency trees able to fully describe any permutation. Permutation trees can handle any possible permutation and are therefore more expressive than ITG (Wu, 1997), which can only produce binarizable permutations. In a permutation tree, constituents are labeled with the permutation of their children.

To induce the reordering grammar from parallel data, Stanojević and Sima'an (2015) define a generative probabilistic model that is estimated using the Expectation Maximization algorithm. As during training only the source sentence and the permutation are observed, Expectation Maximization is a suitable choice to model the latent variables in this model. Specifically, there are two main sources of latent variables. Firstly, the exact permutation tree generating a permutation cannot be observed and a single permutation could have been generated by a potentially exponential number of permutaton trees. The model therefore treats the bracketings of these trees as a latent variable. Secondly, the model allows state splitting of non-terminals similar to latent variable syntactic parsing (Matsuzaki et al., 2005; Petrov et al., 2006; Prescher, 2005).

The probability of the observed permutation π takes into account the latent derivations in the model:

$$P(\pi) = \sum_{\Delta \in \text{PEF}(\pi)} \sum_{d \in \Delta} \prod_{r \in d} P(r), \tag{6.3}$$

where PEF(π) is the set of permutation trees able to generate the permutation π (also called the permutation forest), Δ is a permutation tree, d is a derivation of a permutation tree and r is a production rule. The model can be estimated efficiently using maximum likelihood estimation with the Inside-Outside algorithm (Lari and Young, 1990).

To produce a permutation during decoding, the estimated grammar is used to find the derivation of the permutation tree with the lowest expected cost. With the probability of a derivation d defined as

$$P(d) = \prod_{r \in d} P(r), \tag{6.4}$$

the decoding task is as follows:

$$\hat{d} = \arg\min_{d \in \text{Chart}(\mathbf{s})} \sum_{d' \in \text{Chart}(\mathbf{s})} P(d') \, \text{cost}(d, d'), \tag{6.5}$$

where Chart($\mathbf{s}$) is the chart representing all possible derivations of all possible permutation trees for source sentence $\mathbf{s}$. To speed up inference, two modifications are introduced: Firstly, due to its advantageous properties that enable the usage of efficient dynamic programming for computing minimum Bayes risk (MBR, DeNero et al., 2009), Kendall τ is used as a cost function. Secondly, MBR is computed over 10000 unbiased samples from the chart instead of over the full chart itself. To build the permutation lattice with this model we use the top n permutations with the lowest expected Kendall τ cost.

6.4 Machine Translation with Permutation Lattices

6.4.1 Permutation Lattices

For a sentence $\mathbf{s} \equiv s_1 s_2 ... s_n$, we define a permutation lattice as a direct acyclic graph where every path from the initial state to an accepting state traverses exactly n uniquely labeled transitions. Transitions in the lattice are labeled with pairs in $\{(i, s_i)_{i=1}^n\}$. Each path through the lattice represents an arbitrary permutation of the source sentence's n tokens.

Let Q be the set of states and E be the set of transitions, then every path between any two states $u, v \in Q$ has exactly the same length. We denote with out$^*(x)$ the transitive closure of $x \in Q$, which is the set of states reachable from x. If two nodes u and v are connected, i.e. if $v \in \text{out}^*(u)$, then their distance is $d_v - d_u$, where d_x is x's distance from the initial state. This observation can be used to speed up non-monotone translation with permutation lattices: The set of shortest distances, which has to be precomputed for imposing a distortion limit, can now be computed using the transitive closure in time $O(|Q| \times |E|)$ (Simon, 1988) followed by computing single-source distance in time $O(|Q| + |E|)$ (Mohri, 2002). This allows us to avoid having to use a full-fledged cubic time all-pairs shortest path algorithm.

We produce permutation lattices by compressing the n-best outputs from the pre-ordering models into a minimal deterministic acceptor. Unweighted determinization and minimization are performed using OpenFST (Allauzen et al., 2007). The results of this process are very compact representations that can be decoded efficiently. As an illustration, Figures 6.1 and 6.2 show an English sentence from WMT newstest 2014 preordered for translation into German before (6.1) and after minimization (6.2).[2] Table 6.1 shows the influence of the number of predicted permutations on the lattice sizes for English–German. To measure the quality of the predictions, we compare the permutations to the gold permutation obtained from each sentence's word alignments. Kendall τ distance is used to determine how close a predicted permutation is to the gold permutation. The permutation quality for n permutations in Table 6.1 is the correlation of the best out of the top n permutations.

[2]Example sentence: *The Kluser lights protect cyclists, as well as those travelling by bus and the residents of Bergle*.

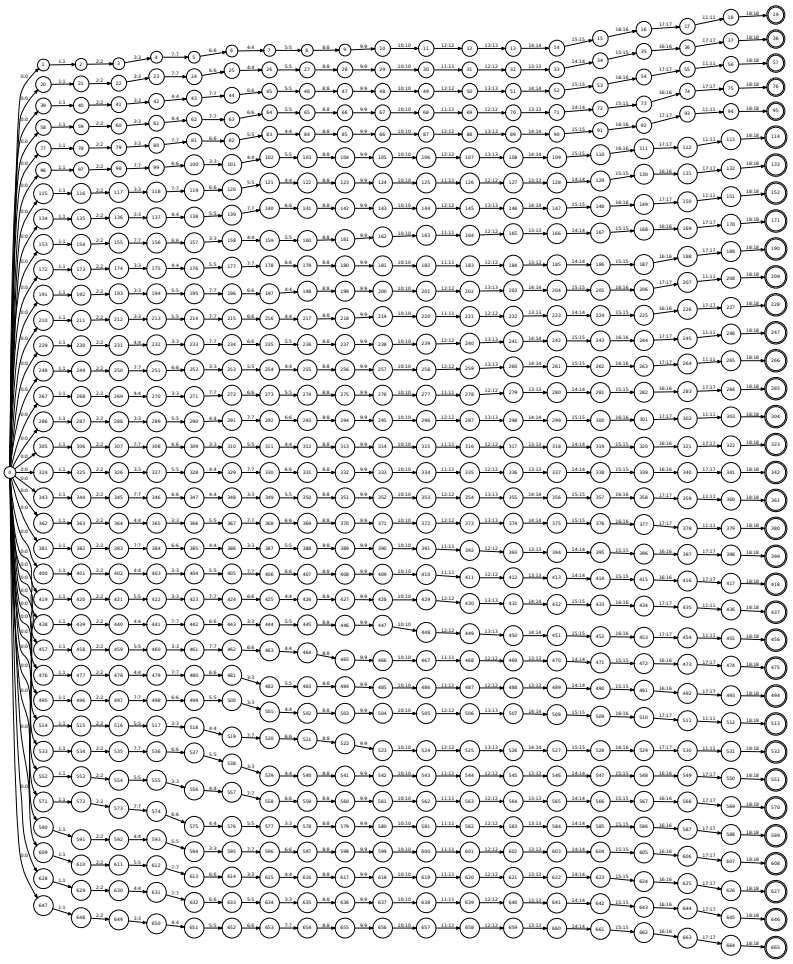


Figure 6.1: Example linear permutation lattice.

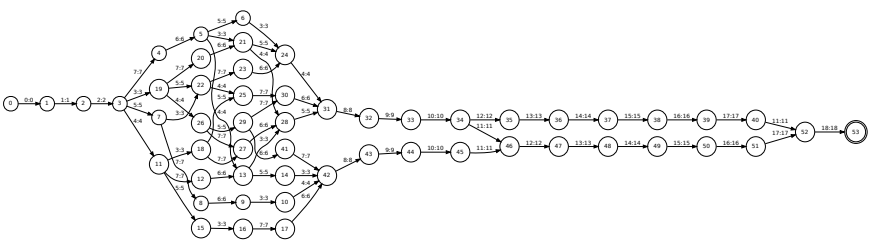


Figure 6.2: Example minimized permutation lattice.

		Lattice	
Permutations	Kendall τ	States	Transitions
Monotone	83.78	23	22
5	84.69	24	52
10	85.23	33	69
100	86.20	72	138
1000	86.75	123	233

Table 6.1: Permutations and lattice size (English–German).

6.4.2 Lattice Silver Training

While for first-best word order predictions, there are two straight-forward options for how to select training instances for the machine translation system, it is less clear how to do this in the case of permutation lattices. In standard preordering, the word order of the source sentence in the training set is commonly determined by reordering the source sentence to minimize the number of crossing alignment links (we denote this as $\mathbf{s}'$). Alternatively, the trained preordering model can be applied to the source side of the training set, which we call $\hat{\mathbf{s}}'_1$. There is a trade-off between both methods: While $\mathbf{s}'$ will generally produce more compact and less noisy phrase tables, it may include phrases that are not reachable by the preordering model. The predicted order $\hat{\mathbf{s}}'_1$, on the other hand, may be too constrained to reach helpful hypotheses (for example, if permutations other than the first-best are preferred). For lattices, one option would be to extract all possible phrases from the lattice directly, but this approach may be noisy, spurious and slow and may result in large phrase tables. Here, we consider a simpler alternative: Instead of selecting either the gold order $\mathbf{s}'$ or the predicted order $\hat{\mathbf{s}}'_1$, we select the order $\hat{\mathbf{s}}'$ which is closest to both the lattice predictions and the gold order $\mathbf{s}'$. Since this order is a mix of the lattice predictions and the gold order, we call this training scheme lattice silver training.

Let $(\mathbf{s}, \mathbf{t})$ be a training instance consisting of a source sentence $\mathbf{s}$ and a target sentence $\mathbf{t}$ and let $\mathbf{s}'$ be the target-order source sentence obtained via the word alignments. For each training instance, we select the preordered source $\hat{\mathbf{s}}'$ as follows:

$$\hat{\mathbf{s}}' = \underset{\hat{\mathbf{s}}'_L \in \pi_k(\mathbf{s})}{\arg\max}\ \text{overlap}(\hat{\mathbf{s}}'_L, \mathbf{s}') \tag{6.6}$$

where $\pi_k(\mathbf{s})$ is the set of k-best permutations predicted by the preordering model. Each $\hat{\mathbf{s}}'_L \in \pi_k(\mathbf{s})$ represents a single path through the lattice. As the cost function, we use n-gram overlap, as commonly used in string kernels (Lodhi et al., 2002):

$$\text{overlap}(\hat{\mathbf{s}}'_L, \mathbf{s}') = \sum_{n=2}^{7} \left(\sum_{c \in C^n_{\mathbf{s}'}} \text{count}_{\hat{\mathbf{s}}'_L}(c) \right), \tag{6.7}$$

where $C^n_{\mathbf{s}'}$ denotes all candidate n-grams of length n in $\mathbf{s}'$ and $\text{count}_{\hat{\mathbf{s}}'_L}(c)$ denotes the number of occurrences of n-gram c in $\hat{\mathbf{s}}'_L$. Ties between permutations with the same overlap are broken using the permutations' scores from the preordering model.

6.5 Experiments

We begin with a description of the experimental setup, datasets and parameters, then describe details of the preordering models and finally present and discuss the results of our experiments.

6.5.1 Experimental Setup

Translation experiments are performed with a phrase-based machine translation system, namely a version of Moses (Koehn et al., 2007) with extended lattice support.[3] We use the basic Moses features and perform 15 iterations of batch MIRA (Cherry and Foster, 2012).

English–Japanese Our experiments are performed on the NTCIR-8 Patent Translation Task (PATMT). Tuning is performed on the NTCIR-7 dev sets, and translation is evaluated on the test set from NTCIR-9. All data is tokenized (using the Moses tokenizer for English and KyTea 5 for Japanese (Neubig et al., 2011)) and filtered to include sentences between 4 and 50 words in length. As a baseline we use a translation system with a distortion limit of 6 and a lexicalized reordering model (Galley and Manning, 2008). We use a 5-gram language model estimated using *lmplz* (Heafield et al., 2013) on the target side of the parallel corpus.

English–German For translation into German, we built a machine translation system based on the WMT 2016 news translation data.[4] The system is trained on all available parallel data, consisting of 4.5m sentence pairs from Europarl (Koehn, 2005), Common Crawl (Smith et al., 2013) and the news commentary corpus. We removed all sentences longer than 80 words and tokenization and truecasing is performed using the standard Moses tokenizer and truecaser. We use a 5-gram Kneser-Ney language model, estimated using *lmplz* (Heafield et al., 2013). The language model is trained on 189m sentences from the target sides of Europarl and news commentary, as well as the News Crawl 2007-2015 corpora. Word alignment is performed using MGIZA++ (using *grow-diag-final-and* symmetrization with 6, 6, 3 and 3 iterations of IBM Model 1, HMM, IBM Model 3 and IBM Model 4). As a baseline we use a translation system with a distortion limit of 6 and a distortion-based reordering model. Tuning is performed on newstest 2014 and we evaluate on newstest 2015.

[3] Made available at `https://github.com/wilkeraziz/mosesdecoder`.

[4] `http://statmt.org/wmt16/`

		Translation	Word order
	DL	BLEU	Kendall τ
Baseline	6	21.76	54.75
Oracle order	6	26.68	58.05
	0	26.41	57.92
First-best	6	21.21[A]	53.44
Lattice (silver)	0	21.88[B]	54.51

[A]Stat. significant against baseline. [B]Stat. significant against first-best.

Table 6.2: Translation results for English–German.

6.5.2 Preordering Models

For German, we use the neural lattice preordering model introduced in Section 6.3.1. The model is trained on the full parallel training data (4.5m sentences) based on the automatic word alignments used by the translation system. Source dependency trees are produced by TurboParser (Martins et al., 2009),[5] which was trained on the English version of HamleDT (Zeman et al., 2012) with content-head dependencies. For translation into Japanese, we train a Reordering Grammar model for 10 iterations of the Expectation Maximization algorithm on a training set consisting of 786k sentence pairs with automatic alignments.

6.5.3 Translation Experiments

We report lowercased BLEU (Papineni et al., 2002) and Kendall τ calculated from the force-aligned hypothesis and reference. Statistical significance tests are performed for the translation scores using the bootstrap resampling method with p-value < 0.05 (Koehn, 2004a). The standard preordering systems ("first-best" in Table 6.2 and 6.4) use an additional lexicalized reordering model (MSD), while the lattice systems use only lattice distortion. For training preordered translation models, we recreate word alignments from the original MGIZA++ alignments and the permutation for English–German and re-align preordered and target sentences for English–Japanese using MGIZA++.[6]

English–German

Translation results for translation into German are shown in Table 6.2. For this language pair, we found standard preordering to work poorly. This is despite the fact that the oracle order (i.e. the source words in the test set are ordered using the word alignments)

[5]`http://cs.cmu.edu/~ark/TurboParser/`

[6]Re-aligning the sentences with MGIZA++ generally improves results, which implies that we are likely underestimating the results for English–German.

shows significant potential. A lattice packed with 1000 permutations on the other hand, performs better even when translating monotonically with a distortion limit of 0.

Lattice Silver Training

To examine the utility of the lattice silver training scheme, we train systems which differ only in the way the training data is extracted. Table 6.3 shows that for English–German, lattice silver training is successful in bridging the gap between the preordering model and the alignment-based target word order, both for monotonic translation and when allowing the decoder to additionally reorder translations.

	Distortion limit	
	0	3
Gold training	21.44	21.60
Lattice silver training	21.88	21.88

Table 6.3: Lattice silver training (BLEU, English–German).

English–Japanese

Results for translation into Japanese are shown in Table 6.4. For this language pair, we found that the first-best preordering approach works well out-of-the-box but providing the translation system with a lattice can improve the results further.

		Translation	Word order
	DL	BLEU	Kendall τ
Baseline	6	29.65	44.87
Oracle order	6	34.22	56.23
	0	30.55	53.98
First-best	6	32.14[A]	49.68
Lattice	0	32.50[AB]	50.79

[A]Stat. significant against baseline. [B]Stat. significant against first-best.

Table 6.4: Translation results for English–Japanese.

6.5.4 Discussion

Although preordering with a single permutation already works well for the strict word order language Japanese, packing the word order ambiguity into a lattice allows the machine translation system to achieve better translation monotonically than allowing a

distortion of 6 and an additional lexicalized reordering model on top of a single permutation. We noticed that lexicalized reordering helped the first-best systems and hence report this stronger baseline. In principle, lexicalized reordering can also be used with lattice translation, and we plan to investigate this option in the future. Linguistic intuition and the empirical results presented in Section 4.4 suggest that compared to Japanese, German shows more word order freedom. Consequently, we assumed that a first-best preordering model would not perform well on the language pair English–German, and indeed the results in Table 6.2 confirm this assumption. For both language pairs, translating a lattice of predicted permutations outperforms the baselines, thus reducing the gap between translation with predicted word order and oracle word order. However, permutation lattices turn out to be the key to enabling even small improvements for the language pair English–German in the context of preordering. This language pair can benefit from the improved interaction between word order and translation decisions. These findings go hand in hand with our analysis in Chapter 4 (see Figures 4.3 and 4.4), specifically the prediction of our information-theoretic word order freedom metric that it should be more difficult to determine German word order from English clues. Our main focus in this chapter was on the language pairs English–German and English–Japanese. Hence, while the results we present in this chapter provide an empirical data point for the utility of permutation lattices for free word order languages, experiments with a broader range of language pairs would provide further empirical support. We perform such preordering experiments with a typologically diverse set of target languages in Chapter 9.

6.6 Conclusion

The world's languages differ widely in how they express meaning, relying on indicators such as word order, intonation or morphological markings. Consequently, some languages exhibit stricter word order than others. Our goal in this part was to examine the effect of word order freedom on machine translation and preordering. We show that addressing uncertainty in word order predictions, and in particular doing so with permutation lattices, can be an indispensable tool for dealing with word order in machine translation. The experiments we performed in this chapter confirm this finding and we further build on it by introducing a new method for training machine translation systems for lattice-preordered input (*lattice silver training*). Finally, we found that while lattices are still helpful for English–Japanese, for which standard preordering already works well, they are crucial for translation into the freer word order language German. In the first part of this thesis, we have explored how typological differences in word order affect machine translation and how these can be addressed. Next to word order, the second major category of typological differences between languages are related to morphology. In the second part of this thesis, we will examine issues caused by varying levels of morphological productivity between languages and will propose methods to bridge such differences.

Part III

Morphological Complexity

Chapter 7

Bridging Typological Differences with Source-Predicted Target Morphology

When translating from a morphologically impoverished to a morphologically rich language, the typological differences of the language pair cause challenges for phrase-based machine translation systems. In this chapter, we examine whether such typological differences can be reduced by enriching the source language with the missing morphological attributes. We present a translation pipeline consisting of two steps: first, the source string is enriched with target morphological features and then fed into a translation model which performs reordering and chooses lexical items matching the provided morphological features. After performing experiments to test the merit of this proposal, we present a model for predicting target morphological features on the source string and its predicate-argument structure and address two major technical challenges: (1) How can we determine which morphological features should be predicted for a specific language pair? and (2) How can predicted morphological features be integrated into the phrase-based model so that it can also be trained on morphological features from the parallel data for a more efficient pipeline? Finally, we evaluate the approach on an English–German translation task and find promising improvement over the baseline phrase-based system.

The content of this chapter is based on the following published article:
Joachim Daiber and Khalil Sima'an. *Machine Translation with Source-Predicted Target Morphology*. In 15th Machine Translation Summit, 2015.
Joachim Daiber performed all experiments and wrote the article. Khalil Sima'an provided guidance and helped editing the article. Khalil Sima'an and Joachim Daiber produced the idea for the article. All chapters of this thesis were written in full by the author.

Chapter Highlights

Problem Statement

- When translating from a morphologically impoverished to a morphologically rich language, the typological differences of the language pair cause challenges for phrase-based machine translation systems: the translation model often lacks access to the signals required for determining the correct target word form, thus leading to data sparsity and impeding the enforcement of morphological agreement over long distances.

Research Question

- Is it possible to bridge typological differences for morphologically improverished source and morphologically rich target languages by enriching the source language with the missing morphological attributes?

Research Contributions

- For translation of a specific language pair, some morphological attributes may be helpful for determining the correct target words while others may be redundant. To select the set of relevant morphological attributes in an efficient and automatic manner, we introduce a latent variable method to select the optimal set of attributes based on the parallel corpus and show that it learns a feature set with quality comparable to a manually selected set for German.
- We introduce a source-side dependency chain model to predict morphological attributes based on source-side syntactic information.
- We explore various ways of integrating the predicted morphological features into the machine translation system and show that it is possible to use predicted features with a translation model trained on morphological features from the parallel data itself, thus enabling a more efficient pipeline.

7.1 Motivation

Typological differences in the means languages employ to express the underlying meaning of a sentence can cause difficulties for machine translation systems. When translating into morphologically rich languages, this poses a challenge for statistical machine translation systems. Rich morphology often co-occurs with relatively freer word order of the target language, making it difficult to predict morphology and word order at the same time. This difficulty is partly due to data sparsity, but morphological agreement

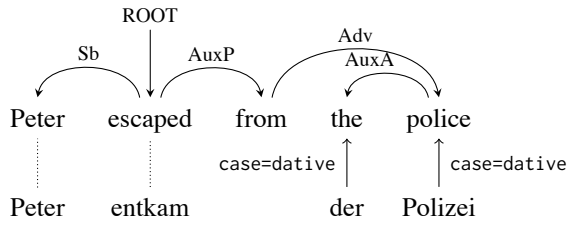


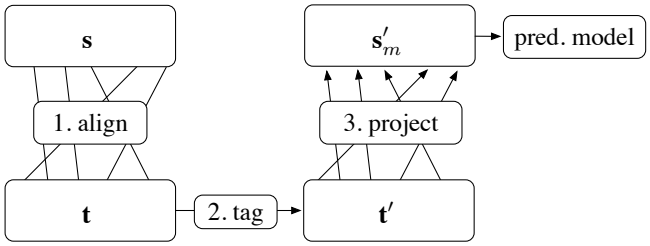
Figure 7.1: Morphology projection from target to source.

between words over long distances also plays a part. In the previous chapters, we have explored techniques to address the challenges caused by word order freedom in such target languages. In this chapter, we explore the idea of combating the sparsity caused by rich target morphology and long-distance agreement by conducting translation in a probabilistic pipeline, in which morphological choice may precede lexical choice and reordering.

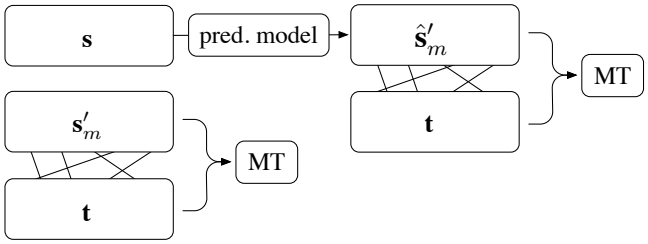
One of the points highlighted in the previous chapters is that in translation, while surface representations might disagree, the predicate-argument structure of a source and a target sentence are often similar. This intuition is the basis for the semantic transfer approaches to machine translation discussed in Section 2.2. When the predicate-argument structure of the source and target sentence are sufficiently similar, we can expect that the linguistic information required for choosing the correct morphological form of the words in the target sentence is present in the source sentence and its syntactic dependency structure. Based on this observation, we explore target morphology as a source-side prediction task which aims at enriching the source sentence with useful target morphological information. Figure 7.1 shows an example illustrating this basic idea. While the German and English sentences both express the same event (Peter escaped from the police), the German sentence uses a dative noun phrase while the English sentence uses a prepositional phrase. In this example, we project the morphological information missing on the source side, namely the grammatical case of the noun phrase (`case=dative`), from the target to the source side.

In practice (see Figure 7.2), after performing word alignment on a sentence pair, we project a subset of the target morphological attributes to the source side via word alignments, and then train a model to predict these attributes on source dependency trees, which we use as a representation of aspects of the source predicate-argument structure. Our approach differs from other approaches to predicting target morphology (e.g. Chahuneau et al., 2013) mainly in that we predict on the source side only. The intuition underlying our approach is similar to the intuition used in the preordering schemes discussed in the previous chapters. While preordering based on source syntax assumes that the source and target syntax are sufficiently similar — an assumption that, as we have illustrated, does not always hold — we only make the weaker assumption that the predicate-argument structures are similar.

We see several technical benefits from predicting target morphology on the source side, which could potentially enable further improvements in machine translation into



(a) Morphology projection and prediction model training.



(b) Machine translation system training.

Figure 7.2: Overview of the training setup and morphology projection.

morphologically rich languages. Source-side prediction models can capitalize on the much reduced complexity of having to represent and process only the input source sentence instead of a large lattice of target hypotheses. Hence, morphological agreement can be enforced over long distances by morphological predictions for the full source sentence. Furthermore, while not pursued in the present work, we hypothesize that the morphological information predicted by our model can be exploited in the word alignment process.

This chapter makes three contributions: Firstly, we report experiments to support the hypothesis that projecting morphology to the source side can be beneficial for translation (Section 7.2), and then present a model for learning to predict target morphology on the source side (Section 7.3). Secondly, we address the question how to automatically learn the set of morphological attributes relevant for a language pair and fitting the parallel training data (Section 7.4). Finally, we introduce methods for integrating this new information into a machine translation system and evaluate on a translation task (Section 7.5).

7.2 Morphology Projection Hypothesis

We are interested in the question whether aspects of target morphology can be directly predicted on the source side. Our hypothesis is that projecting target morphological attributes and learning to predict these on source-side trees can allow the machine translation system to make more informed word form choices. To test this hypothesis, we first

define our representation of morphology and then look to other tasks such as preordering for inspiration on how to perform experiments indicating this approach's potential.

7.2.1 Representation of Morphology

We elect to represent morphology using a common method in natural language processing, namely by associating a set of morphological attributes to each word. Specifically, we use the term *morphological attribute* to refer to any morphological property of a word. Each morphological attribute can assume any of a predetermined set of values, such as {`nom, acc, dat, gen`} for the morphological attribute `case` in the German language. Further, the morphological attributes are refined based on a set of nine atomic parts of speech, yielding a set of morphological attributes of the form `noun:case`, `adj:case`, `verb:tense`, etc.

7.2.2 Testing the Morphology Projection Hypothesis

In other source-side prediction tasks, such as preordering, a common method to measure potential translation improvement is to perform translation experiments with an artificially produced oracle word order, which is obtained using word alignments between the sentences of a development set and their reference translations. Accordingly, to test our hypothesis (i.e., can projecting target morphological attributes to the source side and learning to predict these be beneficial?), we can perform translation experiments using a machine translation setup with and without morphological information projected via word alignments. Our hypothesis can be divided into three questions. The first question is whether projecting target-side morphological attributes to the source side can provide the machine translation system with helpful signals in selecting the correct target words. A second related question is which morphological attributes should be projected. Experiments using word alignments can provide an indication for the potential of this approach, thus allowing us to address the first two questions. They do, however, not answer the third question, namely to which extent target morphology can realistically be predicted on the source side. Hence, in this section we will only focus on the first two questions. The third question will then be addressed in Section 7.3.

We perform translation experiments with translation systems decorated with *projected* morphological attributes. In these systems, the target side of the test set was processed with a morphological tagger and subsets of the resulting morphological attributes were projected to the source side via the word alignments. The translation system is a standard phrase-based machine translation system applied to a training set of several million sentence pairs with a feature-based representation of the morphological attributes. We evaluate translation quality with METEOR and BLEU (Denkowski and Lavie, 2011; Papineni et al., 2002), word order with Kendall τ (Kendall, 1938) and lexical choice with unigram BLEU. Statistical significance is calculated for the translation scores (METEOR and BLEU) using the bootstrap resampling method (Koehn, 2004a). For a more detailed description of the experimental setup, see Section 7.5.3.

		Translation		Word order	Lexical choice
Training & test decoration	Tags	MTR	BLEU	Kendall τ	BLEU-1
None (baseline)	-	35.74	15.12	45.26	49.86
Projected manual set	77	36.34	15.86	45.79	51.30
Projected automatic set	225	36.50	15.73	46.45	51.24
Projected full set	846	36.67	15.96	46.27	51.52

All translation results statistically significant against baseline at $p < 0.01$.

Table 7.1: Translation with various subsets of projected morphology.

These experiments provide a conservative indication of the potential of our approach. They are not oracle translation experiments, but simulate an optimal target morphology prediction model. Results of the experiments are documented in Table 7.1. The three systems differ only in the subset of morphological attributes they use. The results show that projecting target morphological attributes improves translation. Improvements result both from better lexical choice and sometimes also better word order. Using the full set of attributes gives the best METEOR and BLEU scores, but it also contributes significantly to data sparsity. Surprisingly, including only a small, manually selected subset of attributes gives comparable improvement while significantly decreasing the number of resulting tags (combinations of observed morphological features). This *manual* subset is the set of attributes selected for prediction by Fraser et al. (2012), who found that it is beneficial to make some morphological attributes part of the translated word stem instead of predicting them on the target side. The *automatic* selection is a selection of features that our automatic learning procedure, which we will describe in Section 7.4, determined to be the most beneficial for representing the language pair. This selection performed equally well in our experiments.

Hence, while better translation performance is achievable by including all attributes, the prediction task also becomes significantly harder; comparable translation performance can be achieved with a small, well-chosen set of attributes. The good performance of the *manual* set shows that linguistic intuition can be a good starting point for selecting this set; however, a more empirically beneficial set may be selected by enriching the source side only with attributes which help in selecting the correct target words. The fact that the *automatic* set produces a better METEOR score than the *manual* set further supports this intuition.[1] We highlight the METEOR scores here, since for the language pair English–German, METEOR has higher correlation with human judgments than BLEU (Machacek and Bojar, 2014). Now that we established the potential of projecting target morphology to the source side, we aim at capitalizing on this potential. In the next section, we present our model for predicting target morphology on source trees based on source-side dependency chains.

[1]The difference is statistically significant at $p < 0.05$.

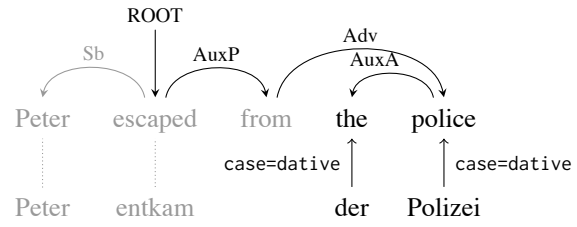


Figure 7.3: Morphology projection and a highlighted source dependency chain.

7.3 Modeling Target-Side Morphology

Since the word order of the source and target language may differ significantly, predicting morphology in a sequential, word-by-word fashion could be inadequate. We assume that the source syntax and the source predicate-argument structure are informative for predicting target morphology. Hence, we propose a source-side dependency chain model, which is expressed as $P(\mathbf{s}'_m \mid \tau, \mathbf{s})$, to predict the morphologically enriched source string $\mathbf{s}'_m$ given a lexical dependency tree τ of the source string $\mathbf{s}$.

7.3.1 Source-Side Dependency Chains

A source-side dependency chain is any path from the root of the source dependency tree to any of its leaf nodes, such as `escaped`→`from`→`police`→`the` in Figure 7.3. Every source node with a 1-to-1 alignment to a target node is decorated with the target node's morphological attributes. A standard morphological tagger, such as the n-th order linear chain conditional random field model (CRF, e.g. Müller et al., 2013), would predict the attribute–value vector for each word left-to-right with a history of $n-1$ tags. Modeling source-side dependency chains instead provides various advantages: Besides providing access to the morphological tags assigned to the dependency tree parent and grandparent nodes, it implicitly encourages morphological agreement between a node and its $n-1$ ancestor nodes. The model also benefits from access to the node's syntactic role, for example to predict grammatical case. Finally, training data sparsity is alleviated because the dependency chain formulation allows the extraction of chains from only partially aligned sentences.

7.3.2 Model Estimation

We estimate the source dependency chain model using the general CRF framework. In a linear-chain CRF model, the probability of a tag sequence $\mathbf{y}$ given a sentence $\mathbf{x}$ is:

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{\exp \sum_{t,i} \lambda_i \cdot \phi_i(\mathbf{y}, \mathbf{x}, t)}{\sum_{\mathbf{y}'} \exp \sum_{t,i} \lambda_i \cdot \phi_i(\mathbf{y}', \mathbf{x}, t)},$$

where t is the index of a token, i is the index of a feature and λ_i is the weight corresponding to the binary feature $\phi_i(\mathbf{y}, \mathbf{x}, t)$. To improve training and inference time, we

		Manual			Automatic			All
		5	6	7	5	6	7	5
Strict	50k	68.50	70.13	68.86	70.84	69.73	70.97	58.33
	100k	67.08	67.38	67.01	69.33	71.15	69.52	58.65
	200k	67.40	67.40	68.55	69.58	69.82	70.06	57.99
Relaxed	50k	72.67	70.36	72.86	**74.67**	71.42	71.83	62.16
	100k	70.01	71.89	69.82	72.63	72.04	72.61	62.18
	200k	69.40	69.46	69.99	71.44	70.80	69.83	60.86

Best overall F_1 score highlighted in bold.

Table 7.2: Impact of attribute selection and model parameters on prediction quality measured by F_1 score.

use a coarse-to-fine pruned CRF tagger (Müller et al., 2013). The training procedure is identical to the linear-chain case, except that we use dependency chains instead of left-to-right chains as training examples. The dependency chain model's feature set is based on the set used in the linear chain CRF for morphological tagging (Müller et al., 2013). Additionally to the features used by Müller et al. (2013), we add the following feature templates: the dependency label of the current token, the dependency label of the parent token, the number of children of the current token, the source-side part-of-speech tag of the token, and the current token's child tokens if they are a determiner (*AuxA*), auxiliary verb (*AuxV*), subject (*Sb*) or a preposition (*AuxP*).

7.3.3 Intrinsic Evaluation

To evaluate the quality of the source dependency chain predictions, we perform experiments on a heldout dataset. Models are trained on a subset of the parallel Europarl data. Evaluation is performed using the F_1 score of the predictions compared to the *projected* morphological attributes obtained by automatic alignment of the source and target side of the evaluation set.

Impact of Model Parameters

Table 7.2 shows prediction performance of the dependency chain model in relation to a selection of model parameters. For each morphological attribute set, we train models of order 5, 6 and 7. All models are trained on sets of 50k, 100k and 200k dependency chains, which are randomly sampled from the training data. In *strict* training mode, we require that target words and source words connected by alignment links agree in their coarse part-of-speech tags. This restriction enforces a weak form of isomorphism between the source and the target sentence and hence limits the training set to training instances of potentially higher quality. In the *relaxed* setup, no such agreement is enforced.

Up to a certain point, higher-order models perform better than models with shorter dependency histories; however, these models are also prone to the issues of data sparsity and overfitting. The results show that strict training performs worse than the relaxed training regime. The strict training regime could possibly produce cleaner training examples; however, since it also enforces a potentially unrealistic isomorphism between the two sentences, those examples may also be less helpful for the final prediction.

Impact of Morphological Attribute Selection

As illustrated in Section 7.2, it is possible to reduce the set of morphological attributes without major losses in translation quality. For the dependency chain model, smaller attribute sets are preferable since they lead to less complex models and faster training. Individual attributes may also be difficult to predict; hence, the exact selection of attributes is important for prediction quality.

	Manual	Automatic	All
Training time, 50k	36m	45m	77m
Training time, 100k	58m	82m	2h51m
Training time, 200k	1h54m	3h5m	6h44m
Tags	77	225	846
Best F_1	72.86	74.67	62.18

Table 7.3: Training times and best scores for the three attribute sets.

Table 7.3 summarizes training times and prediction performance of the three morphological attribute sets. Larger attribute sets and more training examples lead to longer training times. Overall, the automatic set produces more accurate results than the manual selection. Our analysis shows that this is largely due to difficult-to-predict verb attributes, which are included in the manual selection but are not part of the automatically learnt set. The finding that these attributes are hard to predict is in line with Fraser et al. (2012), who equally dropped the prediction of verb attributes in later work.

7.4 Learning Salient Morphological Attributes

Decorating the source language with all morphological properties of the target language would lead to data sparsity and would complicate the prediction task. Therefore, it is necessary to reduce this set to only those morphological attributes which are helpful for a given language pair. We consider a morphological attribute to be salient if it enables the machine translation system to perform better lexical selection. It is computationally infeasible to test all possible combinations of morphological attributes in a full machine translation system; hence, we approximate the machine translation system's ability to perform lexical selection with the word-based translation system defined by

IBM Model 1 (Brown et al., 1993). Based on this simplified translation model, the set of salient features which improve the translation performance can be chosen using a clustering procedure.

7.4.1 Learning Procedure

Let $(\mathbf{s}, \mathbf{t})$ be a pair of parallel sentences in the source and target language. IBM Model 1 provides an iterative method for estimating the translation model $P(\mathbf{t} \mid \mathbf{s})$ from a set of parallel sentences. We add the morphological decoration $\mathbf{s}'_m$ to this model. The translation model now takes the following form:

$$P(\mathbf{t} \mid \mathbf{s}) = \sum_{\mathbf{s}'_m \in \Theta_m(\mathbf{s})} P(\mathbf{s}'_m \mid \mathbf{s}) P(\mathbf{t} \mid \mathbf{s}'_m),$$

where $P(\mathbf{t} \mid \mathbf{s}'_m)$ is the standard IBM Model 1 formulation applied to morphologically decorated source tokens. In this simple machine translation model, the morphological attributes are directly concatenated to the source words. For example, if the English token `police` is decorated with grammatical case, gender and number, it would be replaced by the string `police/case=dat+gender=female+number=singular`. We define the log-likelihood of a set of parallel sentences $\mathbf{X}$ to be:

$$\mathcal{L}(\mathbf{X}) \equiv \log \prod_{(\mathbf{s},\mathbf{t}) \in \mathbf{X}} P(\mathbf{t} \mid \mathbf{s}) P(\mathbf{s}) = \sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{X}} \log P(\mathbf{t} \mid \mathbf{s}) + \log P(\mathbf{s})$$

Let M_0 be the initial set of all morphological attributes observed in the training corpus. Our goal is to find the set $M_n \subseteq M_0$ which maximizes the likelihood of a heldout dataset. An alternative to choosing a subset of M_0 would be to learn a latent representation directly; however, since using a subset of M_0 makes the resulting selection interpretable and since it simplifies the subsequent training of the prediction model, we opt for using the subset approach. By ${\mathbf{s}'_m}^{(i)}$ we denote the decorated source sentence containing only the morphological attributes in M_i. We formulate the search for the set M_n as follows:

$$\begin{aligned}
M_n &= \arg\max_{M_i \subseteq M_0} \sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{X}} \log P(\mathbf{t} \mid \mathbf{s}) + \log P(\mathbf{s}) \\
&= \arg\max_{M_i \subseteq M_0} \sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{X}} \log P(\mathbf{t} \mid \mathbf{s}) \\
&= \arg\max_{M_i \subseteq M_0} \sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{X}} \log \left(\sum_{\mathbf{s}'_m \in \Theta_m(\mathbf{s})} P(\mathbf{s}'_m \mid \mathbf{s}) P(\mathbf{t} \mid {\mathbf{s}'_m}^{(i)}) \right)
\end{aligned}$$

We found the estimates for $P(\mathbf{s}'_m \mid \mathbf{s})$ using the full set of attributes M_0 to be reasonable, with sufficient probability mass assigned to the most likely path. Therefore, we

approximate this model by only using the first-best (Viterbi) assignment $\mathbf{s}''_m$. The final, simplified search objective is therefore:

$$\begin{aligned} M_n &= \arg\max_{M_i \subseteq M_0} \sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{X}} \log \Big(P(\mathbf{s}''_m \mid \mathbf{s}) P(\mathbf{t} \mid \mathbf{s}''^{\,(i)}_m) \Big) \\ &= \arg\max_{M_i \subseteq M_0} \sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{X}} \log P(\mathbf{t} \mid \mathbf{s}''^{\,(i)}_m) \end{aligned}$$

The optimal set of attributes can now be determined with a clustering procedure starting from the full set of morphological attributes M_0. This procedure is reminiscent of Petrov et al. (2006) since as in their work, we can simulate the removal of a morphological attribute by merging the statistics of each of its occurrences: To simulate the removal of the attribute `gender`, for example, we would merge the statistics of every occurrence of the attribute (either `gender=male` or `gender=female`). The two tags `case=nom+gender=female` and `case=nom+gender=male` would therefore be merged into one tag `case=nom`.

In summary, the procedure is as follows:

1. Initialization:
   - Estimate the source dependency chain model $P(\mathbf{s}'^{\,(0)}_m \mid \mathbf{s})$, apply it to decorate the training and heldout set, producing $\mathbf{T}_0$ and $\mathbf{H}_0$ (datasets $\mathbf{T}$ and $\mathbf{H}$ decorated with M_0).
   - Estimate $P(\mathbf{t} \mid \mathbf{s}''^{\,(0)}_m)$: perform 5 iterations of IBM Model 1 training on $\mathbf{T}_0$.
2. Start with $i = 0$.
3. Calculate $P(\mathbf{t} \mid \mathbf{s}''^{\,(i)}_m)$ for each sentence pair in the heldout set $\mathbf{H}_i$.
4. Find the attribute $\hat{m} \in M_i$, such that:

   $$\hat{m} = \arg\min_{m\prime \in M_i} \Bigg(\sum_{(\mathbf{s},\mathbf{t}) \in \mathbf{H}_i} \log P(\mathbf{t} \mid \mathbf{s}''^{\,(i)}_m) - \log P(\mathbf{t} \mid \mathbf{s}''^{\,(i)\setminus m'}_m) \Bigg),$$

   where $\mathbf{s}''^{\,(i)\setminus m'}_m$ denotes a sentence with the attributes in M_i minus attribute m'.
5. Merge all values of $\hat{m}$ in $\mathbf{T}_i$ and $\mathbf{H}_i$, producing $\mathbf{T}_{i+1}$ and $\mathbf{H}_{i+1}$.
6. Estimate $P(\mathbf{t} \mid \mathbf{s}''^{\,(i+1)}_m)$: Merge the t-tables containing $\hat{m}$ and perform IBM Model 1 iteration on $\mathbf{T}_{i+1}$.
7. Repeat from (3) with $i = i + 1$. Stop if no possible merge improves $\mathcal{L}(\mathbf{H}_i)$.

7.4.2 Intrinsic Evaluation

The complexity of the clustering procedure is $O(|M| \times k \times l^2)$ for k sentences of length l. In practice, the procedure runs several hours on a standard personal computer. Table 7.4 shows the attributes determined by the learning procedure. The column *Auto* shows the procedure's selection and the column *Manual* shows the manually determined set of morphological attributes for the same language pair, as used by Fraser et al. (2012).

Noun		Adjective		Verb		Other	
Manual	Auto	Manual	Auto	Manual	Auto	Manual	Auto
gender†	gender	gender†	gender	number‡*	-	-	part:neg
number	number	number‡	number	person‡*			part:sbpos
case	case	case‡	case	tense*			punc:type
				mode*			num:type
		declension	synpos				
			degree				

† Transferred with lemma. ‡ Propagated from noun. * Dropped in later work.

Table 7.4: Salient attributes for English–German.

Quality of the Selection

From inspection of these attributes, we find that our method learns a reasonable set of salient attributes. The manual and automatic selections differ mainly in the verb attributes, which our learning procedure removed from the final set. Morphological attributes in the manual selection marked with † are attributes that in the work of Fraser et al. (2012) were transferred as part of the translated stem by their machine translation system. The symbol ‡ marks morphological attributes that they propagated from the noun (for example, an adjective's case is copied from the noun it modifies). Finally, the verb attributes, which are marked with *, are used by Fraser et al. (2012) but found to be problematic by Cap et al. (2014b) and dropped in later work (Cap et al., 2014a). Likewise, inspection of our model showed that verb attributes perform badly as they may be difficult to predict. Hence, our procedure successfully learnt not to model these attributes while retaining the beneficial noun and adjective attributes.

Granularity of the Morphological Attributes

When simulating the removal of a morphological attribute with this learning algorithm, all of its values are merged. In some language pairs, however, it may be useful to merge the individual values of the attributes instead. For example, from the spelling of German nouns it is usually not recognizable whether the noun is `case=nominative` or `case=accusative`. Hence, the algorithm should ideally be able to also merge individual values. Since this is a straight-forward extension of our current algorithm, we plan to evaluate this aspect in future work.

7.5 Morphology-Informed Machine Translation

To leverage the morphology predictions in a machine translation decoder, we integrate this additional information into the translation model. During training and tuning, the translation model is decorated with morphological attributes either projected from the target side or predicted by our dependency chain model.

7.5.1 Integration of Target Morphology Predictions

In practice, the predicted morphological attributes on the source side can be integrated into the machine translation system as arbitrary features based on source morphology and target strings. In our experiments, we opted for a feature representation in which this information is encoded as source morphology-to-target affix features. We chose this simple representation because it is generic enough as a representation on the one hand and it is not prone to overfitting on the other hand. For each phrase candidate on the source side, sparse features fire for a given sequence of source-side morphology tags and target-side string affixes. As an example, consider the sentence *Peter entkam der Polizei* (Peter escaped from the police) from Figure 7.3. In this case, the morphological attributes gender (*female*), number (*singular*) and grammatical case (*dative*) would have been projected from the target to the source side for the phrase *the police*/*der Polizei*. When translating the source segment *the police*, the feature `gender=fem+number=sing+case=dat X` $\rightarrow$ `-er X` would fire based on the predicted morphology. This hint would help the machine translation system choose the correct German determiner *der*.[2]

7.5.2 Inference Strategies

At test time, the morphological decoration of the source sentence needs to be selected. This decision should ideally take into account both the predictions of our source-side dependency chain model and the content of the phrase table, which may be decorated with projected morphology.

We compare several inference strategies. The major distinction between these strategies is whether the machine translation system is trained and tuned on projected morphology or predicted morphology. Training on predicted morphology has the benefit that it lets the machine translation system learn how much it can trust the predictions made by the dependency chain model. However, this method is also more laborious in system development, since it requires retraining and tuning the entire translation system for every change in the prediction model.

Training and Decoding with Viterbi Predictions

In the first decoding setup, which is similar to the most common setup used in preordering, we decorate both the training and the test set with the Viterbi decorations extracted from the dependency chain model. Specifically, for each possible dependency chain in the source dependency tree, we preform standard CRF Viterbi tagging starting from the root of the tree. The full training and tuning set is decorated with these single-best predicted decorations. System training and tuning is then performed on these predictions.

[2]This feature example is taken from the weights of the system trained with the automatic morphological attribute set and predicted training and test decoration.

During test time, only the single-best Viterbi prediction is considered by the machine translation system.

Training on Projected Morphology and Decoding with Viterbi Predictions

The "projected" training setup differs from the previous setup in that the morphological decorations on the training and tuning set are not predicted but projected from the target side via the word alignments. During test time, the decorations are predicted using single-best Viterbi predictions as in the previous setup. While this strategy is advantageous since it simplifies the system training, its main downside is that it cannot take into account possible shortcomings of the prediction model. At training time, only projected decorations are observed, which might not be realistic when taking into account the prediction model.

7.5.3 Evaluation

Having introduced and evaluated the attribute selection process and the prediction of target-side morphological attributes based on source-side dependency chains, we now turn to the evaluation of the predicted morphological information within a full machine translation pipeline.

Experimental Setup

We use a phrase-based machine translation system (Cer et al., 2010) with a 5-gram language model and distortion-based reordering (dl=5). Features based on the source morphology predictions are learnt on either the projected morphology or the predictions of the source dependency chain model. Experiments are conducted on English–German. Source-side dependency trees are predicted based on the HamleDT treebank (Zeman et al., 2012) using TurboParser (Martins et al., 2010). The dependency parser is trained to produce pseudo-projective dependency trees (Nivre and Nilsson, 2005).[3] The system is trained on the full parallel sections of Europarl (Koehn, 2005) and tuned and tested on the WMT 2009 and WMT 2010 newstest sets respectively.

Monolingual morphological tagging is performed using the Marmot CRF-based tagger (Müller et al., 2013). The tagger is trained on the English and German parts of the HamleDT treebank. The morphological attributes of both languages follow the Interset standard (Zeman, 2008), which contains 45 unique attribute vectors (tags) for English and 958 for German.

Discussion

Table 7.5 shows the outcomes of using the inference strategies presented in Section 7.5.2. We evaluate translation quality with METEOR and BLEU (Denkowski and Lavie,

[3] Projectivization was performed using MaltParser version 1.8; `http://www.maltparser.org/`.

		Translation		Word order	Lexical choice
Attributes	Training decoration	MTR	BLEU	Kendall τ	BLEU-1
None	-	35.74	15.12	45.26	49.86
Manual	Predicted	35.85	15.19	45.43	50.01
	Projected	34.63[A]	14.00[A]	44.07	48.75
Automatic	Predicted	35.99[AC]	15.23[B]	45.88	50.27
	Projected	35.98[AC]	15.22[C]	45.89	50.27

[A]Statistically significant against baseline at $p < 0.05$ [B]Statistically significant against baseline at $p < 0.06$
[C]Statistically significant against Manual selection at $p < 0.05$

Table 7.5: Translation with predicted test decorations.

2011; Papineni et al., 2002), word order with Kendall τ (Kendall, 1938) and lexical choice with unigram BLEU. Statistical significance tests are performed for the translation scores using the bootstrap resampling method (Koehn, 2004a).

The results show that both attribute selections show improvements over the baseline when training and testing on predicted morphology. On the other hand, when training on projected morphology and performing Viterbi predictions, a visible gap between the manual set and the automatic set can be observed. This gap indicates that with the automatic set, the predictions by the dependency chain model are closer to the projected predictions so that the machine translation system learns realistic weights for the prediction part. Additionally, the system based on the automatic selection produces a significantly better METEOR score than the system using the manual selection. As in the experiments with projected morphology, the results of this evaluation indicate that the improvements stem from both word order choices as well as better lexical selection. In terms of time performance, we found that the additional information does not significantly affect the speed of the translation system. The Viterbi algorithm for predicting the target morphology is efficient and as the information is passed to the machine translation system as sparse features, no additional complexity is added. While we have focused on the language pair English–German, the methods presented in this chapter are applicable to many other language pairs. We therefore aim to perform additional experiments for morphologically rich target languages such as Turkish, Arabic and Czech in future work.

7.6 Related Work

Various approaches have been proposed to the problem of translating between languages of varying morphological complexity. Avramidis and Koehn (2008) enrich the morphologically impoverished source side with syntactic information and translate via

a factored machine translation model. The work of Avramidis and Koehn (2008) is closely related to the present work; however, while their decorations are source-side syntactic information (e.g. whether a noun is the subject), we directly predict target morphology and learn to select the most relevant properties automatically. A similar approach, in which source syntax is reduced to part-of-speech tags, is used successfully for translation into Turkish (Yeniterzi and Oflazer, 2010). Following the tradition of two step machine translation (Bojar and Kos, 2010), Fraser et al. (2012) translate morphologically underspecified tokens and add inflections on the target side based on the predictions of discriminative classifiers.

Carpuat and Wu (2007), Jeong et al. (2010), Toutanova et al. (2008) and Chahuneau et al. (2013) propose discriminative lexicon models that are able to take into account the larger context of the source sentence when making lexical choices on the target side. These proposals differ mostly in the way that the additional morphological information is integrated into the machine translation process. Jeong et al. (2010) integrate their lexical selection model via features in the underlying treelet translation system (Quirk et al., 2005). Toutanova et al. (2008) survey two basic methods of integration. In the first method, the inflection prediction model is allowed to change the inflections produced by the underlying machine translation system. The second method is a two step method, where the machine translation system translates into target-language stems, which are then inflected by the inflection model. Chahuneau et al. (2013) create *synthetic phrases*, i.e. phrases with inflections that have not been observed directly in the training corpus but have been created by an inflection model. These synthetic phrases are then added to the training data of the machine translation system and marked as such. This in turn enables the machine translation system to learn how much to trust them. Finally, Williams and Koehn (2011) add unification-based constraints to the target side of a string-to-tree model. The constraints are extracted heuristically from a treebank and violations of these constraints are then penalized during decoding.

7.7 Conclusion

In this chapter, we have explored the novel approach of target morphology projection. After testing the idea empirically, we have proposed three components to realize this idea: First, we introduced the dependency chain model for predicting arbitrary target morphology attributes based on source dependency trees. Second, we introduced a learning procedure to determine a language pair's set of salient morphological attributes. And finally, we have introduced and compared various strategies for integrating this new information into a machine translation system. The experiments we have performed have provided several insights: They have demonstrated that projecting a small subset of morphological attributes to the source side can provide major translation improvements while reducing the complexity of prediction. Furthermore, our approach for learning this useful subset performs well based on both intrinsic evaluation and the empirical results during prediction and translation. Given that previous

work has found it rather difficult to achieve improvements in German morphology, we consider the improvements in METEOR score and the modest improvements in BLEU score encouraging. Apart from morphological inflection, a second area of word formation leading to great typological differences between languages and causing issues for statistical machine translation is compounding. In the next chapter, we will examine this problem based on the language pair German–English and propose and evaluate an unsupervised method for splitting compound words into their meaningful parts.

Chapter 8

Aligning Word Formation Processes: A Semantic Approach to Compound Splitting

Compounding is a highly productive word formation process in some languages that is often problematic for natural language processing applications. In this chapter, we investigate whether distributional semantics in the form of word embeddings can enable more semantically motivated processing of compounds than standard string-based methods. We present an unsupervised approach that exploits regularities in the semantic vector space (based on analogies such as "bookshop is to shop as bookshelf is to shelf") to produce compound analyses of high quality. A subsequent compound splitting algorithm based on these analyses is highly effective, particularly for ambiguous compounds. German–English machine translation experiments show that this semantic analogy-based compound splitter leads to better translations than a commonly used frequency-based method.

The content of this chapter is based on the following published article:
Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. *Splitting Compounds by Semantic Analogy*. In 1st Deep Machine Translation Workshop, 2015.

Joachim Daiber and Stella Frank produced the initial research idea and provided guidance to Lautaro Quiroz and Roger Wechsler. Stella Frank additionally wrote an early draft of Section 4.2. Joachim Daiber additionally helped with running experiments and wrote the article. Lautaro Quiroz and Roger Wechsler developed initial versions of the code, ran experiments and wrote early drafts of several sections. All chapters of this thesis were written in full by the author.

Chapter Highlights

Problem Statement

- Phrase-based models combine minimal units, most commonly words, into longer phrases. Highly productive word formation processes such as compounding pose a challenge since they produce new words while obscuring from the translation system the minimal units these new words are composed of.

Research Question

- Can semantic analogy based on distributional semantics in the form of word embeddings be used to split compound words into their components?

Research Contributions

- We show that regularities in the semantic vector space (based on analogies such as "bookshop is to shop as bookshelf is to shelf") can be used to produce compound analyses of high quality.
- We develop a compound splitting algorithm based on these analyses and show that it is highly effective on a German–English machine translation task.

8.1 Motivation

In languages such as German, compound words are a frequent occurrence leading to difficulties for natural language processing applications, and in particular machine translation. Several methods for dealing with this issue — from shallow count-based methods to deeper but more complex neural network-based processing methods — have been proposed. The recent surge in practical models for distributional semantics has enabled a multitude of practical applications in many areas, most recently in morphological analysis (Soricut and Och, 2015). In this chapter, we investigate whether similar methods can be utilized to perform more semantic processing of compounds. A great asset of word embeddings are the regularities that their multi-dimensional vector space exhibits. Mikolov et al. (2013) showed that regularities such as "*king* is to *man* what *queen* is to *woman*" can be expressed and exploited in the form of basic linear algebra operations on the vectors produced by their method. This often-cited example can be expressed as follows: $v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$, where $v(\,\cdot\,)$ maps a word into its word embedding in vector space.

Soricut and Och (2015) exploit these regularities for unsupervised morphology induction. Their method induces vector representations for basic morphological transformations in a fully unsupervised manner. String prefix and suffix replacement rules are

induced directly from the data based on the idea that morphological processes can be modeled on the basis of *prototype* transformations, i.e. vectors that are good examples of a morphological process are applied to a word vector to retrieve its inflected form. A simple example of this idea is $\uparrow d_{cars} = v(\text{cars}) - v(\text{car})$ and $v(\text{dogs}) \approx v(\text{dog}) + \uparrow d_{cars}$, which expresses that the relation of the word *car* to the word *cars* is the same as the relation of the word *dog* to the word *dogs*. The direction vector $\uparrow d_{cars}$ represents the process of adding the plural morpheme *-s* to a noun.

While this intuition works well for frequently occurring inflectional morphology, it is not clear whether it extends to more semantically motivated word formation processes such as compounding. We study this question in the present chapter. Our experiments are based on the German language, in which compounding is a highly productive phenomenon allowing for a potentially infinite number of combinations of words into compounds. This fact, coupled with the issue that many compounds are observed infrequently in data, leads to a data sparsity problem that complicates the processing of such languages. Our contributions are as follows: First, we study whether the regularities exhibited by the vector space also apply to compounds (Section 8.2). We examine the relationship between the components within compounds, as illustrated by the analogical relationship "*Hauptziel* is to *Ziel* what *Hauptader* is to *Ader*."[1] By leveraging this analogy we can then analyze the novel compound *Hauptmann* (captain) by searching for known string prefixes (e.g. *Haupt-*) and testing whether the resulting split compound *Haupt|mann* has a similar relation between its components (*haupt*, *mann*) as the prototypical example *Haupt|ziel*. We induce the compound components and their prototypes and apply them in a greedy compound splitting algorithm, which we evaluate on a gold standard compound splitting task (Section 8.3) and as a preprocessing step in a machine translation setup (Section 8.4).

8.2 Compounds and Morphology Induction

Our approach is based on the work of Soricut and Och (2015), who exploit regularities in the vector space to induce morphological transformations.

8.2.1 Morphology Induction from Word Vectors

Soricut and Och (2015) extract morphological transformations in the form of prefix and suffix replacement rules up to a maximum length of 6 characters. The method requires an initial candidate set containing all possible prefix and suffix rules that occur in the monolingual corpus. For English, the candidate set contains rules such as `suffix:ed:ing`, which represents the suffix *ed* replaced by *ing* (e.g. *walked→walking*).

[1] In vector algebra: $\uparrow d_{\text{Hauptziel}} = v(\text{Hauptziel}) - v(\text{Ziel})$ and $v(\text{Hauptader}) \approx v(\text{Ader}) + \uparrow d_{\text{Hauptziel}}$. The compounds translate to "main goal" (*Hauptziel*) and "main artery" (*Hauptader*). As a separate noun, *Haupt* means head.

This candidate set also contains over-generated rules that do not reflect actual morphological transformations; for example `prefix:S:`ϵ[2] in *Scream→cream*.

The goal is to filter the initial candidate set to remove spurious rules while keeping useful rules. For all word pairs a rule applies to, word embeddings are used to calculate a vector representing the transformation. For example, the direction vector for the rule `suffix:ing:ed` based on the pair (*walking*, *walked*) would be $\uparrow d_{walk\underline{\text{ing}\rightarrow\text{ed}}} = v(\textit{walked}) - v(\textit{walking})$. For each rule there are thus potentially as many direction vectors as word pairs it applies to. A direction vector is considered to be meaning-preserving if it successfully predicts the affix replacements of other, similar word pairs. Specifically, each direction vector is applied to the first word in the other pair and an ordered list of suggested words is produced. For example, the direction vector $\uparrow d_{walk\underline{\text{ing}\rightarrow\text{ed}}}$ can be evaluated against (*playing*, *played*) by applying $\uparrow d_{walk\underline{\text{ing}\rightarrow\text{ed}}}$ to *playing* to produce the predicted word form: $v(\textit{played}*) = v(\textit{playing}) + \uparrow d_{walk\underline{\text{ing}\rightarrow\text{ed}}}$. This prediction is then compared against the original word embedding $v(\textit{played})$ using an evaluation function $E(v(\textit{played}), v(\textit{playing}) + \uparrow d_{walk\underline{\text{ing}\rightarrow\text{ed}}})$.[3] If the evaluation function passes a certain threshold, we say that the direction vector *explains* the word pair. Some direction vectors explain many word pairs while others explain few. To judge the explanatory power of a direction vector, a *hit rate* metric is calculated, expressing the percentage of applicable word pairs for which the vector makes good predictions.[4] Each direction vector has a hit rate and a set of word pairs that it explains (its evidence set). Apart from their varying explanatory power, morphological transformation rules can also be ambiguous. For example, the rule `suffix:`ϵ`:s` can describe both the pluralization of a noun (one *house*→two *houses*) and the 3rd person singular form of a verb (I *find*→she *finds*). Different direction vectors may explain the nouns and verbs separately.

Soricut and Och (2015) retain only the most explanatory vectors by applying a recursive procedure to find the minimal set of direction vectors explaining most word pairs. We call this set of direction vectors *prototypes*, as they represent a prototypical transformation for a rule and other words are formed *in analogy to* this particular word pair. Intuitively, it may seem surprising that the vector space of word embeddings can be used to learn representations of morphological processes, since this vector space is only informed by the other words directly surrounding a word. Nevertheless, Soricut and Och (2015) show that their prototypes can be applied successfully in a word similarity task for several languages, including morphologically rich languages such as Arabic. In this chapter, we use word embeddings to analyze the relationship between compound words and their components, which is more markedly a semantic relationship and should therefore be more robustly covered by word embeddings.

[2] ϵ denotes the empty string.

[3] We follow Soricut and Och (2015) in defining E as either the *cosine* distance or the *rank* (position in the predictions).

[4] A transformation is considered a *hit* if the evaluated score is above a certain threshold for each evaluation function E.

8.2.2 Compounds and the Semantic Vector Space

Compounds can be classified into several groups (Lieber and Štekauer, 2009): in *endocentric* compounds, the semantic head is part of the compound (a birdhouse is a also a house) and in *exocentric* compounds the semantic head is outside of the compound (a skinhead is not a head). In this chapter, we focus on endocentric compounds, which are also the most frequent type in German (Dressler et al., 2012, p. 254). Endocentric compounds consist of a modifier and a semantic head. The semantic head specifies the basic meaning of the word and the modifier restricts this meaning. In German, the modifiers come before the semantic head; hence, the semantic head is always the last component in the compound. When applying the idea of modeling morphological processes by semantic analogy to compounds, we can represent either the semantic head or the modifier of the compound as the transformation (like the morpheme rules above). Since the head carries the compound's basic meaning in endocentric compounds, we add the modifier's vector representation to the head word in order to restrict its meaning. We expect the resulting compound to be in the neighborhood of the head word in the semantic space (*birdhouse* should be close to *house*).

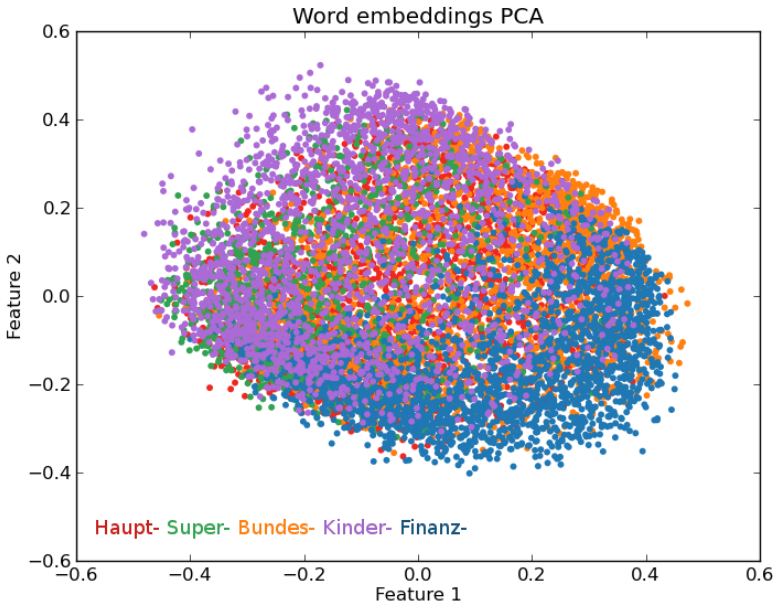
We illustrate this intuition by visualizing compound words and their components in the vector space. All visualizations are produced by performing principal component analysis (PCA) to reduce the vector space from 500 to 2 dimensions. Figure 8.1 presents the visualization of various compounds with either the same head or the same modifier. For Figure 8.1a, we plot all German compounds in our dataset that have one of the modifiers *Haupt-*,[5a] *Super-, Bundes-*,[5b] *Kinder-*[5c] or *Finanz-*.[5d] Figure 8.1b plots all German compounds with one of the heads *-arbeit*,[5e] *-ministerium*,[5f] *-mann*[5g] or *-stadt*.[5h] Hence, the two plots illustrate the difference between learning vector representations for compound modifiers or heads. Words with the same modifier do not necessarily appear in close proximity in vector space. This is even less likely for modifiers that can be applied liberally to many head words, such as *Super-* or *Kinder-*.[5c] On the other hand, compounds with the same head are close in the embedding space. This observation is crucial to our method, as we aim to find direction vectors that generalize to as many word pairs as possible.

8.3 Compound Induction from Word Embeddings

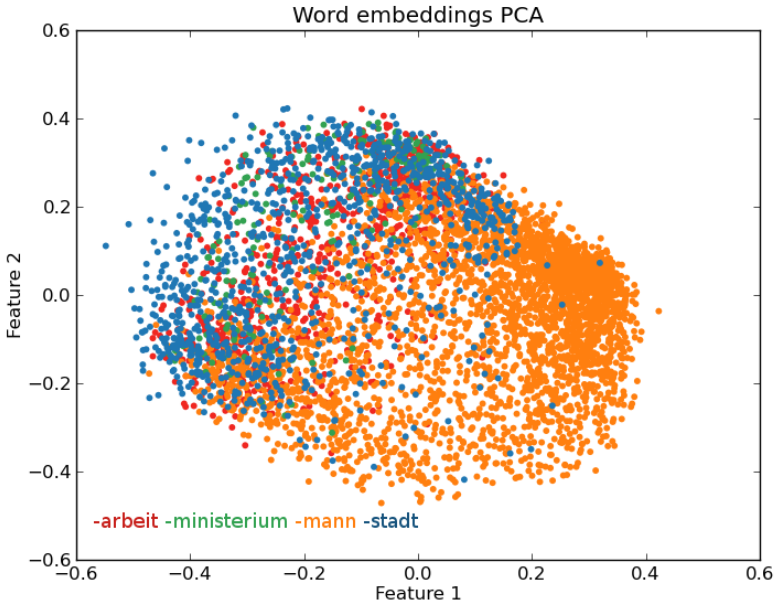
8.3.1 Extracting Candidates

The initial set of modifier candidates is produced from all possible string prefixes of 4 or more characters (for languages in which the semantic head precedes the modifier, we

[5]Gloss for modifiers: (a) main, (b) federal, (c) children, (d) finance. Heads: (e) piece of work, (f) ministry, (g) man, (h) city.



(a) Compounds with the same modifier.



(b) Compounds with the same head.

Figure 8.1: Semantic representations of compounds based on (a) their modifiers and (b) their semantic heads.

	Modifier	Support		Modifier	Support
1.	*Land-*	8387	6.	*Landes-*	5189
2.	*Kinder-*	6249	7.	*Schul-*	5011
3.	*Haupt-*	5855	8.	*Jugend-*	4855
4.	*Lande-*	5637	9.	*Ober-*	4799
5.	*Stadt-*	5327	10.	*Groß-*	4656

Table 8.1: Modifiers by size of support set.

would extract suffixes instead).[6] We retain a modifier as a candidate if both the modifier and the rest of the word, which is the potential head, occur in our vocabulary. The initial candidate set contains 281k modifiers, which are reduced to 165k candidates by removing the modifiers occurring in only one word. The length of the average support set (i.e., the set of all compounds the modifier applies to) is 13.5 words. Table 8.1 shows the ten candidate modifiers with the biggest support sets. At this stage, the candidate set contains any modifier-head split that can be observed in the data, including candidates that do not reflect real compound splits.[7] Compound splits are not applied recursively here, as we assume that internal splits can be learnt from the occurrences of the heads as individual words.[8]

8.3.2 Extracting Prototypes

To find the prototype vectors that generalize best over the most words in the support set, we apply the recursive procedure of Soricut and Och (2015). The algorithm initially computes the direction vector for each (*modifier*, *compound*) pair in the support set by performing a subtraction of the head embedding and the compound embedding, e.g. $\uparrow d_{\underline{\text{dog}}\textit{house}} = v(\textit{doghouse}) - v(\textit{house})$.

Each direction vector is then evaluated by applying it to all the word pairs in the support set, for example $v(\textit{owner}) + \uparrow d_{\underline{\text{dog}}\textit{house}} \stackrel{?}{=} v(\textit{dogowner})$ for the word pair *dog*|*owner*. If the resulting vector is close (according to E) to the vector of the target compound, we add it to the evidence set of the vector. The direction vector with the largest evidence set is selected as a prototype. All pairs this prototype explains are then removed and the algorithm is applied recursively until no direction vector explains at least t_{evd} compounds. As the evaluation function E we use the rank of the correct word in the list of predictions and perform experiments with $t_{\text{evd}} = \{10, 6, 4\}$. Lastly, to ensure efficient computation, we down-sample the evidence set to 500 words.

[6] For efficient computation, we use a directed acyclic word graph: `https://pypi.python.org/pypi/pyDAWG`.

[7] For example, as *Para* (a river) and *dies* (this) occur in the data, an incorrect candidate split occurs for *Para*|*dies* (paradise).

[8] For example, for *Haupt*|*bahn*|*hof* (main train station), we observe both *Haupt*|*bahnhof* and *Bahn*|*hof* in the data.

Prototype	Evidence words
$v_{\text{-Zeiger}}$	-Bewegung -Klicks -Klick -Tasten -Zeiger
$v_{\text{-Stämme}}$	-Mutanten -Gene -Hirnen -Stämme
$v_{\text{-Kostüm}}$	-Knopf -Hirn -Hirns -Kostüm
$v_{\text{-Steuerung}}$	-Ersatz -Bedienung -Steuerung

Table 8.2: Prototypes and evidence words for *Maus-*.[11]

8.3.3 Implementation and Intrinsic Evaluation

We now turn to implementation considerations and perform an intrinsic evaluation of the prototypes.

Word embeddings As monolingual data, we use the German parts of the *News Crawl Corpus* (2007-2014).[9] The text is truecased and tokenized, and all punctuation characters are removed, resulting in approximately 2 billion tokens and a vocabulary size of 3 million words. We use *word2vec* to estimate the word embeddings using the skip-gram model.[10] Embeddings are trained with a window size of 5, using 500 dimensions and a minimum word frequency of 2. The relatively low minimum frequency threshold means that word embeddings are created even for infrequent words, thus ensuring that word embeddings are created for complex compound words, which may occur very few times in the data. While producing word embeddings for very rare words poses the risk that their representations may be of poor quality, we found this to not be an issue in our experiments.

Treatment of interfixes (Fugenelemente) For mostly phonetic reasons, German allows the insertion of a limited set of characters between the modifier and the head. As learning this set is not the aim of our work, we only allow the fixed set of interfixes {*-s-*, *-es-*} to occur. We add all combinations of interfix and case variations of the head word to the modifier's support set.

What do the prototypes encode? An inspection of the prototypes for each modifier shows that the differences between them are not always clear-cut. Often, however, each prototype expresses one specific sense of the modifier. Table 8.2 illustrates this observation using the German modifier *Maus-* as an example. The German word *Maus*, *mouse* in English, can refer to both the computer device and the animal. Although there are more than two prototype vectors, it is interesting to observe that the two senses are almost fully separated.

[9] `http://www.statmt.org/wmt15/translation-task.html`

[10] `https://code.google.com/p/word2vec/`

[11] Words are related to mouse pointer (*Zeiger*), biological genus (*Stämme*), mouse costume (*Kostüm*) and control (*Steuerung*).

	(a) Mean hit rate		(b) Mean cosine similarity	
$t_{\text{rank}} =$	80	100	80	100
$t_{\text{evd}} = 4$	26%	22%	0.39	0.39
$t_{\text{evd}} = 6$	31%	26%	0.43	0.43
$t_{\text{evd}} = 10$	36%	31%	0.45	0.45

	(c) % with prototypes		(d) Mean number of prototypes	
$t_{\text{rank}} =$	80	100	80	100
$t_{\text{evd}} = 4$	8.93%	9.52%	4.20	4.16
$t_{\text{evd}} = 6$	5.13%	5.47%	3.29	3.30
$t_{\text{evd}} = 10$	2.91%	3.14%	2.25	2.29

Table 8.3: Overview of the influence of hyper-parameters on prototype extraction.

Quality of the prototypes To evaluate the quality of our extracted prototypes, we use the hit rate metric defined by Soricut and Och (2015). A direction vector's hit rate is the percentage of relevant word pairs that can be explained by the vector. A prediction is explainable if the correct target word is in the top t_{rank} predictions and, optionally, if there is a cosine similarity of at least t_{sim}. The implementation of this evaluation function E requires the calculation of the cosine distance between a newly created vector and the word vector of every item in the vocabulary. Since this score is calculated N times for every of the N word pairs (i.e., N^2 times), this is an extremely computationally expensive process. For more efficient computation, we use an approximate k-nearest neighbor search method.[12] While this is not a lossless search method, it offers an adjustable trade-off between the model's prediction accuracy and running time.[13] For our standard setting ($t_{\text{evd}} = 6, t_{\text{rank}} = 80$), the hit rates using approximate and exact rank are 85.9% and 60.9% respectively. This shows that the approximate method produces hit rates that are more optimistic, which will affect how the prototype vectors are extracted. Additionally, restricting both *rank* and *similarity* ($t_{\text{rank}} = 80, t_{\text{sim}} \geq 0.5$) leads to lower hit rates (25.9% for approximate and 15% for exact rank).

Influence of the thresholds Table 8.3 compares the parameters of our model based on (a) the mean hit rate, (b) cosine similarity, (c) the percentage of candidate modifiers with at least one prototype and (d) the mean number of prototypes per rule. Higher values of t_{evd} (minimum evidence set size) lead to better quality in terms of hit rate and cosine similarity as prototypes have to be able to cover a larger number of word pairs in order to be retained. The rank threshold t_{rank} also behaves as expected. Reducing t_{rank} to 80 produces predicted vectors of higher quality since they have to be closer to

[12] `https://github.com/spotify/annoy`

[13] With this fast approximate search method the total training time would be just below 7 days if run on a single 16 core machine.

the compound embeddings. Tables (c) and (d) illustrate that using a more restrictive parameter setting reduces the number of modifiers for which a prototype can be extracted. Of a total of 165k candidate prefixes, only 3%-10% are not filtered out using these settings. The average number of prototypes per modifier also decreases with more restrictive settings. Interestingly, however, for the most restrictive setting ($t_{\text{evd}} = 10$, $t_{\text{rank}} = 80$), this number is still a relatively high 2 prototypes per vector.

8.3.4 Compound Splitting

To obtain a clearer view of the quality of the extracted compound representations, we apply the prototypes to a compound splitting task.

Splitting Compounds by Semantic Analogy

The extracted compound modifiers and their prototypes can be employed directly to split a compound into its components. Algorithm 3 presents the greedy algorithm, which can be applied to individual words in a text. *V* is the word embedding vocabulary, *M* is the set of extracted modifiers with their prototypes, and PREFIXES(·) is a function returning all string prefixes.

Algorithm 3 Greedy compound splitting algorithm.

```
1  procedure DECOMPOUND(word, V, M)
2      modifiers ← {m | p ← PREFIXES(word) if p ∈ M}
3      if modifiers = ∅ OR word ∉ V then
4          return word
5      bestModifier ← ∅
6      for modifier ∈ modifiers do
7          head ← word without modifier        ▷ e.g. house ← doghouse without dog-
8          if head ∈ V then
9              for (head_proto, word_proto) ∈ modifier do
10                 Evaluate "word is to head what word_proto is to head_proto"
11                                 ▷ e.g. doghouse is to house what dogowner is to owner
12                 Update bestModifier if this is the best match so far
13     return word split based on bestModifier
```

Compounds may only be split if (a) the full compound word is in the vocabulary *V*, i.e. it has been observed at least twice in the training data (Line 3), (b) it has a string prefix in the modifier set and this modifier has at least one prototype (Line 3), (c) the potential head word resulting from splitting the compound based on the modifier is also in our vocabulary (Line 8). The last case, namely that the compound head candidate is not in the vocabulary can occur for two reasons: either this potential head is a valid word that has not been observed frequently enough or, the more common occurrence,

	This work		Moses (partial)		Moses (full)	
Scenario	Accuracy	Coverage	Accuracy	Coverage	Accuracy	Coverage
Full test set	27.43	58.45	18.04	31.41	6.57	13.75
2 splits	24.94	56.75	13.13	20.13	1.79	3.11
3 splits	21.10	68.37	8.04	18.35	1.21	2.92
4 splits	22.09	62.11	9.98	15.91	1.19	1.90
5 splits	24.04	69.23	9.62	11.54	0.96	1.92

Table 8.4: Evaluation of all compounds and highly ambiguous compounds only.

the substring does not form a valid word of the language.[14] The algorithm's coverage can be increased by backing off to a frequency-based method if conditions (a) or (c) are violated. The core of the algorithm is the evaluation of meaning preservation in Line 10. This evaluation is performed using the *rank*-based and *cosine similarity*-based evaluation functions. Modifiers that do not pass the thresholds defined for these functions are discarded as weak splits. To split compounds with more than two components, the algorithm is applied recursively.

Evaluation of Compound Splitting

We use the test set from Henrich and Hinrichs (2011), which contains 54k compounds, each annotated with binary splits. We require a minimal prefix length of 4 characters and, hence, also filter the test set accordingly. This step leaves 50651 compounds for evaluation. Moses (Koehn et al., 2007) offers a compound splitter that splits a word if the geometric average of the frequencies of its components is higher than the frequency of the full compound. We use two instances of this compound splitter as baselines: one using the German monolingual dataset used to train the word2vec models and a second using a subset of the previous dataset.[15] Results for the full test set (accuracy and coverage, i.e. $\frac{|\text{correct splits}|}{|\text{compounds}|}$ and $\frac{|\text{compounds split}|}{|\text{compounds}|}$) are presented in the first row of Table 8.4.

Splitting highly ambiguous compounds The semantic nature of our approach enables us to discriminate good candidate splits from bad ones. By capturing the meaning relation between compounds and their components, the method can disambiguate between the various splitting rules for each compound. Contexts where multiple split points apply to a compound should therefore be handled particularly well by our approach. We simulate different ambiguity scenarios based on the Henrich and Hinrichs (2011) gold standard dataset: We extract compounds with 2, 3, 4, or 5 potential split

[14] For example, when applying the algorithm to *Herrengarderobe* (male cloak room), two possible prefixes can be extracted: The prefixes *Herr* and *Herren*. For *Herr*, the remaining substring is *engarderobe*, which is by itself not a German word. Hence, in this case the candidate prefix can be discarded.

[15] Subset: *News Crawl 07-09* (275m tokens, 2.09m types). Full: *07-14* (2b tokens, 3m types).

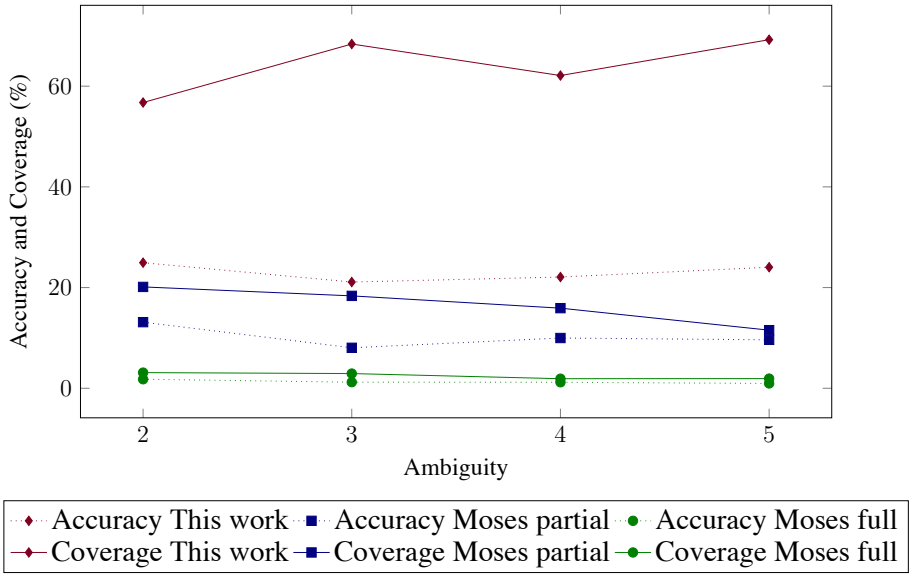


Figure 8.2: Evaluation of highly ambiguous compounds.

points.[16] The resulting test sets consists of 18571, 1815, 842 and 104 compounds, respectively. For all compound splitting experiments, prototype vectors are extracted using $t_{\text{evd}} = 6$ and $t_{\text{rank}} = 100$.

Table 8.4 presents accuracy and coverage for the compounds within the different ambiguity scenarios. To better visualize the trends for highly ambiguous compounds, we plot the accuracy and coverage scores in relation to the ambiguity of the compounds in Figure 8.2. The analogy-based method outperforms the frequency-based baselines in both coverage and accuracy. While for the Moses splitter, the coverage decreases with increasing ambiguity, the opposite behavior is exhibited by our approach. Having more possible splits produces a larger number of direction vectors, which in turn increases the likelihood of obtaining a meaning-preserving split. This experiment shows that the analogy-based compound splitter is advantageous for words that can potentially be explained by several candidate splits.

8.4 Compound Splitting for Machine Translation

8.4.1 Translation Setup

We use the Moses decoder (Koehn et al., 2007) to train a phrase-based machine translation system on the German–English *Common Crawl* parallel corpus and *WMT news*

[16]Each string prefix that occurs as a separate word produces a potential split (indicated by }). Potential split points may not be linguistically motivated and can lead to correct (*general|stabs*) or incorrect splits (*gene}rals}tabs*). Examples include Einkauf}s}wagen, Eis}en}bahn}unternehmen, Wissen}s}chaft}s}park and Gene}ra}l}s}tab}s.

	(a) No compound splitting			(b) OOV only		
	Splits	BLEU	MTR	Splits	BLEU	MTR
Moses splitter	0	17.6	25.5	226	17.6	25.7[A]
This work				317	17.6	25.8[A]

	(c) Rare: c(w) < 20			(d) All words		
	Splits	BLEU	MTR	Splits	BLEU	MTR
Moses splitter	231	17.6	25.7	244	17.9	25.8[A]
This work	744	18.2[ABC]	26.1[ABC]	1616	17.7	26.3[A]

[A] Statistically significant against (a) at $p < 0.05$ [B] Statistically significant against Moses splitter at same c(w) at $p < 0.05$
[C] Statistically significant against best Moses splitter (d) at $p < 0.05$.

Table 8.5: Translation results for various integration methods.

test 2010 (tuning). Word alignment is performed with GIZA++ (Och and Ney, 2003). We use a 4-gram language model estimated using IRSTLM (Federico et al., 2008), as well as lexicalized reordering. The test data set is *WMT news test* 2015,[17] which contains approximately 2100 German–English sentence pairs and 10000 tokens (using one reference translation). We compare our method against a translation system without any handling of compounds, and the same system using Moses' default compound splitter, an implementation of the frequency-based compound splitter discussed above. The test set contains 2111 out-of-vocabulary word types (natural OOV words), yielding 2765 unknown tokens, consisting mostly of compounds, brand and city names. This implies that 22.16% of the word types and 7.15% of the tokens of the test corpus are out-of-vocabulary words for the baseline system.

8.4.2 Translation Experiments and Discussion

To test our compound splitter in a realistic setting, we perform a standard machine translation task. We translate a German text using a translation baseline system without compound handling, system (a) in Table 8.5, a translation system integrating the Moses compound splitter trained using the best-performing settings, and a translation system using our analogy-based compound splitter. We test the following basic methods of integration: Splitting only words that are out-of-vocabulary to the translation model, system (b) in Table 8.5, splitting all words that occur less than 20 times in the training corpus, system (c) in Table 8.5, and applying the compound splitters to every word in the datasets, system (d) in Table 8.5. Table 8.5 shows the results of these translation experiments. For each experiment, we report BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), and the number of compound splits performed on the test set. Statistical significance is calculated using bootstrap resampling (Koehn, 2004a).

[17] `http://www.statmt.org/wmt15/translation-task.html`

The results show that when applied without restrictions, our method splits a large number of words and leads to minor improvements. When applied only to rare words the splitter produces statistically significant improvements in both BLEU and METEOR over the best frequency-based compound splitter. This difference indicates that a better method for deciding which words the splitter should be applied to could lead to further improvements. Overall, the output of the analogy-based compound splitter is more beneficial to the machine translation system than the baseline splitter.

8.5 Related Work

8.5.1 Splitting Compounds for Machine Translation

Highly productive word formation such as compounding can be a cause for data sparsity in statistical machine translation. When translating from a compounding language, two main problems have to be addressed from a technical point of view: Firstly, to split compound words into their components potential split points have to be disambiguated. Secondly, it has to be decided whether a compound should be split at all. When translating into a compounding language, the individual components of a compound word have to be merged into a single word. We only address translation from a compounding language in this chapter, for approaches to dealing with compound merging, see Fraser et al. (2012) and Cap et al. (2014a).

One of the most straight-forward approaches to compound splitting was presented by Koehn and Knight (2003), who split compounds based on their components' frequency. Apart from this frequency-based algorithm, Koehn and Knight (2003) also present more complex approaches using word alignments and part-of-speech tags; however, while these more advanced approaches provide better intrinsic performance (measured against hand-annotated segmentations), the more basic frequency-based approach results in the best translation quality. This discrepancy is likely caused by the fact that phrase-based systems do not necessarily penalize the over-splitting of compounds, since the components of a compound can be handled as a phrase. Employing more resource-intensive tools, Nießen and Ney (2000), Popović et al. (2006) and Fritzinger and Fraser (2010) use morphological analyzers to split German compound words. Combined with frequency information, these methods can provide improved intrinsic performance and translation quality. As these approaches rely on supervised morphological analyzers, they are orthogonal to our approach, which is fully unsupervised.

Another avenue of research has dealt with the question of whether it is beneficial to only split compositional compounds while taking care not to split lexicalized compounds. Weller et al. (2014) use this approach by only splitting words that pass a threshold of distributional similarity to its components. When applying the resulting split compounds in a phrase-based machine translation system, they find that, similarly the observation of Koehn and Knight (2003), over-splitting does not pose a problem for the translation system and hence not splitting lexical compounds does not improve

translation quality. Our approach also relies on distributional semantics for determining similarity. However, the approach we follow in this chapter is fully unsupervised, requiring only word embeddings estimated from a monolingual corpus. Additionally, our approach is significantly less complex, making it easy to understand and implement.

8.5.2 Semantic Compositionality

Noun compounds have also played a role in the field of distributional semantics itself. Reddy et al. (2011) find that distributional properties can explain the relationship between English compound components and the full compound, a result that was later replicated by Schulte im Walde et al. (2013) for German compounds. Schulte im Walde et al. (2013) further studied whether syntactically motivated word embeddings would help in this task but found them to be inferior to standard word vectors.

8.6 Conclusion

In this chapter, we have studied whether regularities in the semantic word embedding space can be exploited to model the composition of compound words based on analogy. To approach this question, we made the following contributions: First, we evaluated whether properties of compounds can be found in the semantic vector space. We found that this space lends itself to modeling compounds based on their semantic head. Based on this finding, we discussed how to extract compound transformations and prototypes following the method of Soricut and Och (2015) and proposed an algorithm for applying these structures to compound splitting. Our experiments show that the analogy-based compound splitter outperforms a commonly used compound splitter on a gold standard task. Our novel compound splitter is particularly adept at splitting highly ambiguous compounds. Finally, we applied the analogy-based compound splitter in a machine translation task and found that it compares favorably to a commonly used shallow frequency-based method. In the first two parts of this thesis, we have examined how typological differences between languages influence machine translation and have proposed various methods to address them. In the next chapter, we turn to the question of whether we can, beyond solely addressing typological differences, exploit our knowledge about the typological properties of languages to build more robust, universal models for machine translation.

Part IV

Linguistic Typology as a Knowledge Source

Chapter 9

Universal Reordering via Linguistic Typology

In the previous chapters, we observed that typological properties of the languages targeted in natural language processing tasks influence performance and we have examined various methods to address this issue. In this chapter, we examine how linguistic typology itself can be used as a rich source of information in machine translation. In particular, we explore the idea of building a universal reordering model from English to a large number of target languages. To build this model, we exploit typological features of word order for a large number of target languages together with source (English) syntactic features. We train a single model on a combined parallel corpus representing all (22) involved language pairs. Apart from empirically demonstrating the value provided by typological descriptions of language, our proposed method can produce word order predictions for a broad range of languages, including language pairs with little or no parallel data. When the universal reordering model is used for preordering followed by monotone translation (no reordering inside the decoder), our experiments show that this pipeline gives comparable or improved translation performance with a phrase-based baseline for a large number of language pairs (12 out of 22) from diverse language families.

The content of this chapter is based on the following published article:
Joachim Daiber, Miloš Stanojević, and Khalil Sima'an. *Universal Reordering via Linguistic Typology*. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics.

Joachim Daiber performed experiments and wrote the article. Miloš Stanojević and Joachim Daiber produced the idea for the article. All authors edited the article. All chapters of this thesis were written in full by the author.

Chapter Highlights

Problem Statement

- If the similarities and differences between languages can be captured with a small set of parameters, as various linguistic theories suggest, then models of word order should be able to benefit by gaining better generalization and requiring less training data for individual languages. Such benefits are not attainable when training reordering models for each language pair individually, but a universal reordering model applicable to and trained on a range of typologically diverse languages could take advantage of them.

Research Question

- Can linguistic typology serve as a source of knowledge to guide reordering models and to facilitate universal reordering models applicable to multiple target languages?

Research Contributions

- We show that typological information collected in the WALS database, when combined with neural network techniques, can be used to build a universal reordering model.
- We evaluate this model in a translation task from English into various typologically diverse languages and show that it provides comparable or improved translation performance to a phrase-based baseline for a large number of languages.

9.1 Motivation

As we have seen in previous chapters, various linguistic theories and typological studies suggest that languages often share a number of properties and that their differences fall into a small set of parameter settings (Chomsky, 1965; Greenberg, 1966a; Comrie, 1981). In Chapter 3.2, we have provided an overview of how these theories explain the differences and commonalities between languages, and we have examined and addressed issues in machine translation caused by typological differences in word order freedom (Part II) and morphological complexity (Part III). While intuitions about language universals and language typology have influenced work on multilingual parsing (Zeman and Resnik, 2008; McDonald et al., 2011), linguistic typology has found less practical use in guiding or informing models in other areas of natural language processing, such as the task of machine translation. A main area where typological differences

between languages cause difficulties, and an area which we have highlighted in this thesis, is differences in word order. These differences are frequently given special treatment, such as in the case of preordering. The preordering approach works well for some language pairs, however it usually demands a separate, dedicated preordering model for every source-target language pair, trained on a word-aligned corpus specific to the particular language pair.

However, if the similarities and differences between languages can indeed be captured with a small set of parameters, as various linguistic theories suggest, then models of word order should try to benefit from the similarities between target languages in the training data. This benefit is not attainable when training a separate preordering model for every new target language. Hence, ideally, the word-aligned data obtained for various target languages should be combined to train a single universal reordering model with a single set of parameters. In this chapter, we address two fundamental questions: (1) can a linguistically inspired universal reordering model show promising experimental results and (2) how can such a universal reordering model be built?

For building an effective universal reordering model, we require access to a resource that describes the similarities and differences between (target) languages using a small set of properties. The World Atlas of Language Structures (Dryer and Haspelmath, 2013), WALS, is a major resource which currently specifies the abstract linguistic properties of 2679 languages.[1] In this chapter we explore the use of the linguistically defined features in WALS for a broad range of target languages and show that these features have merit for building a single, universal reordering model. The universal reordering model is based on a feed-forward neural network which is trained to predict the target word order given source syntactic structure and all available WALS parameter settings for each of the 22 target languages involved. By training the feed-forward neural network on the WALS-enriched data from a broad range of target languages, we enable the universal reordering model to both learn how much to trust the WALS parameters and to exploit possible interactions between them for different target languages. When the universal reordering model is followed by *monotone translation* (no reordering inside the decoder), our experiments show that this pipeline gives comparable or improved translation performance to a Moses baseline with a distortion limit of 6 for a large number of language pairs. This suggests that typological target language features could play a key role in building better, more general preordering models, which have, until now, been trained solely on source sentences and word alignments, but had no access to other target-side information.

We believe that the experiments presented in this chapter have both theoretical and practical implications. Firstly, they show the utility and provide empirical support for the value of linguistic typology in machine translation. Secondly, they enable building more compact preordering models that should generalize to a broad set of target languages and which potentially apply for the low resource setting where little or no parallel data is available for a specific target language.

[1] For more uses of WALS in natural language processing, see Section 3.2.4 of Chapter 3.

9.2 Related Work

The most basic usage of linguistic knowledge in preordering is in restricting the search space of possible word order choices by using syntactic parse trees. Earlier work was done mostly on constituency trees (Khalilov and Sima'an, 2012; Xia and McCord, 2004) while more recent versions of preordering models mostly use dependency trees (Lerner and Petrov, 2013; Jehl et al., 2014). Preordering in syntax-based models (whether dependency or constituency) is done on the local level where for each constituent (or head word in dependency-based models) the classifier decides how the children (or dependent words) should be reordered.

Employing classifiers to make local decisions on each tree node is one machine learning approach to solving this problem. An alternative to employing machine learning techniques is the direct use of linguistic knowledge, which can in some cases give clear rules for the reordering of children in the tree. An early example of rule-based preordering is by Collins et al. (2005), who develop linguistically justified rules for preordering German into English word order. Similar in spirit but significantly simpler is the approach of Isozaki et al. (2010), who exploit the fact that Japanese word order is in large part the mirror image of English word order — the heads of constituents in English are in final position while they are in initial position in Japanese. Preordering English sentences into Japanese word order thus only involves two simple steps: (1) Finding the parse tree of the English sentence (the authors used HPSG derivations) and (2) moving the head of each constituent to the initial position. However, this approach does not seem to scale easily because manually encoding reordering rules for all the world's language pairs would be a rather difficult and very slow process.

In contrast to manually encoding rules for language pairs, we could use similarities and differences between target languages encoded in existing *typological databases* of structural properties of the world's languages, such as the World Atlas of Language Structures (Dryer and Haspelmath, 2013). The challenge we address in this chapter is how to exploit typological databases such as WALS to guide the learning algorithm into making the right decisions about word order. So if, for instance, a feature indicates that the target language follows VSO (verb-subject-object) word order, then the preordering algorithm should learn to transform the English parse tree from SVO into VSO order. Using typological features like these in a machine learning system for preordering constitutes a compromise between knowledge-based (rules) and data-driven (learning) approaches to preordering.

9.3 Linguistic Typology as a Knowledge Source

The field of linguistic typology, as introduced in Chapter 3.2, studies the similarities and distinguishing features between languages and aims at classifying them accordingly. Among other areas, the World Atlas of Language Structures describes general properties of each language's word order. Overall, WALS contains 192 features, but

not all features are relevant to determining word order. Many WALS features deal with phonology, morphology or lexical choice: Feature 129A, for example, describes whether the language's words for "hand" and "arm" are the same. Hence, for simplicity's sake we pre-select the subset of WALS features potentially relevant to determining word order and describe this subset in the following. Table 9.1 provides an overview of these features, along with an indication of the relative frequency distribution of each of their values over all languages in WALS.

In Section 3.2.3 we have discussed the various means of classification of languages in linguistic typology. Traditionally, the most common classification is according to the order of the subject, the object and the verb in a transitive clause. Accordingly, a number of WALS features describe the order of these elements. WALS Feature 81A classifies languages into 6 dominant clause-level word orders. For languages such as German or Dutch, which do not exhibit a single dominant clause-level order, Feature 81B describes 5 combinations of two acceptable word orders. Additionally, two features describe whether the verb precedes the subject (82A) and whether the verb precedes the object (83A). The position of adjuncts in relation to the object and the verb are described in Feature 84A and the internal structure of adpositional phrases is described in Feature 85A, which specifies whether the language uses pre-, post- or inpositions. Finally, the following properties describe the order of words in relation to nouns: Feature 86A specifies the position of genitives (e.g. "the girl's cat"), Feature 87A the position of adjectives (e.g. "yellow house"), Feature 89A the position of numerals (e.g. "10 houses") and Feature 90A the position of relative clauses (e.g. "the book that I am reading") in relation to the noun.

9.4 Universal Reordering Model

Our universal reordering model uses a preordering architecture similar to the (non-universal) preordering model of De Gispert et al. (2015), which in turn is based on the authors' earlier work on logistic regression and graph search for preordering (Jehl et al., 2014).

9.4.1 Basic Preordering Model

In this neural preordering model, a feed-forward neural network is trained to estimate the swap probabilities of nodes in the source-side dependency tree. The learning task is built around the following question: How likely is it that two nodes *a* and *b* are in the linear order (a, b) or (b, a) in the target language? Preordering then consists of finding the best sequence of swaps according to this model. While De Gispert et al. (2015) use a depth-first branch-and-bound algorithm to find the best permutation, we use the k-best version of this algorithm and minimize the resulting preordering finite-state automaton to produce a lattice of word order choices (see Daiber et al., 2016a, or Chapter 6 of this thesis).

WALS feature		Observed feature values #	Distribution
37A	Definite Articles	5	
46A	Indefinite Pronouns	5	
48A	Person Marking on Adpositions	4	
52A	Comitatives and Instrumentals	3	
53A	Ordinal Numerals	8	
54A	Distributive Numerals	7	
55A	Numeral Classifiers	3	
56A	Conjunctions and Universal Quantifiers	3	
57A	Position of Pronominal Possessive Affixes	4	
61A	Adjectives Without Nouns	7	
66A	The Past Tense	4	
67A	The Future Tense	2	
68A	The Perfect	4	
69A	Position of Tense-Aspect Affixes	5	
81A	Order of Subject, Object and Verb	7	
81B	Two Dominant SVO Orders	5	
82A	Order of Subject and Verb	3	
83A	Order of Object and Verb	3	
84A	Order of Object, Oblique, and Verb	6	
85A	Order of Adposition and NP	5	
86A	Order of Genitive and Noun	3	
87A	Order of Adjective and Noun	4	
88A	Order of Demonstrative and Noun	6	
89A	Order of Numeral and Noun	4	
90A	Order of Relative Clause and Noun	7	
91A	Order of Degree Word and Adjective	3	
92A	Position of Polar Question Particles	6	
93A	Position of Interrogative Phrases	3	
94A	Order of Adverbial Subordinator and Clause	5	
95A	Rel. between Order of O + V and Adp. + NP	5	
96A	Rel. between Order of O + V and Rel. Cl. + N	5	
97A	Rel. between Order of O + V and Adj. + N	5	

Table 9.1: WALS features potentially relevant to determining word order.

Model Estimation

Training examples are extracted from all possible pairs of children of the source dependency tree node, including the head itself. The crossing score of two nodes a and b (a precedes b in linear order) is defined based on each node's set of aligned target indexes (A_a and A_b) as follows:

$$\text{cs}(a, b) = |\{(i, j) \in A_a \times A_b : i > j\}| \tag{9.1}$$

A pair of nodes (a, b) is swapped if $\text{cs}(b, a) < \text{cs}(a, b)$, i.e. if swapping reduces the number of crossing alignment links. Training instances generated in this manner are then used to estimate the order probability $p(i, j)$ for two indexes i and j. The best possible permutation of each node's children (including the head) is determined via graph search. The score of a permutation π of length k consists of the order probabilities of all possible pairs:

$$\text{score}(\pi) = \prod_{1 \leq i < j \leq k | \pi[i] > \pi[j]} p(i, j) \prod_{1 \leq i < j \leq k | \pi[i] < \pi[j]} 1 - p(i, j) \tag{9.2}$$

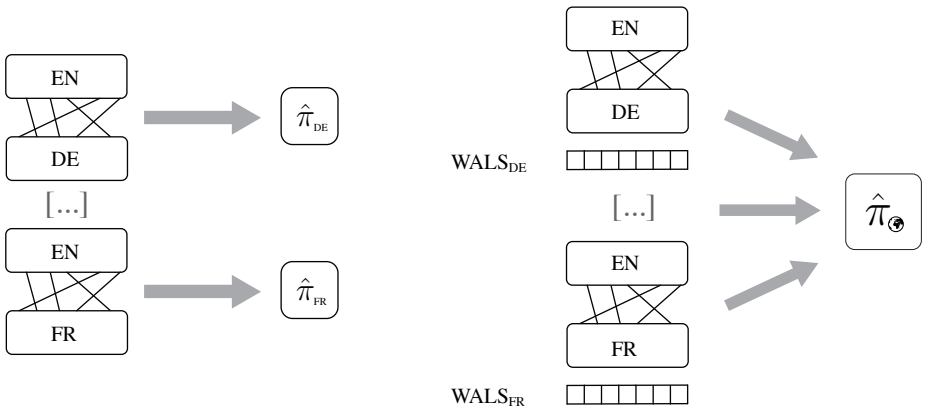
De Gispert et al. (2015) use a feed-forward neural network (Bengio et al., 2003) to predict the orientation of a and b based on 20 source features, such as the words, POS tags, dependency labels, etc.[2]

Permutation Lattices

In order to find the sequence of swaps leading to the best overall permutation according to the model, the score of a permutation is obtained by extending a partial permutation π' of length k' by one index i (Jehl et al., 2014). This score can be efficiently computed as:

$$\text{score}(\pi' \cdot \langle i \rangle) = \text{score}(\pi') \prod_{j \in V | i > j} p(i, j) \prod_{j \in V | i < j} 1 - p(i, j) \tag{9.3}$$

Instead of extracting the single-best permutation, we use the k-best extension of branch-and-bound search (van der Poort et al., 1999). The resulting k-best permutations are then compressed into a minimal deterministic acceptor and unweighted determinization and minimization are performed using OpenFST (Allauzen et al., 2007). For a more detailed description of the preordering model and permutation lattices, please see Chapter 6 of this thesis.



(a) Basic preordering model. (b) Universal reordering model.

Figure 9.1: Training of basic preordering models and the universal reordering model.

9.4.2 Estimating a Universal Reordering Model

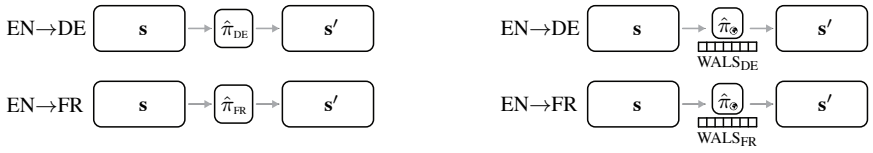
The universal reordering model differs from the basic neural preordering model in terms of its features and in how the training data is collected. Differences in training data collection are illustrated in Figure 9.1. Figure 9.2 shows how the model is applied.

In addition to the source features used in the standard neural preordering model (cf. Section 9.4.1), we add a feature indicating the source word order of the two tokens, as well as the type of end-of-sentence punctuation. We then add WALS features 37, 46, 48, 52–57, 61, 66–69 and 81–97. In our model, WALS features are represented by their ID and the value for the current target language (e.g. "WALS_87A=Adj-Noun" or "WALS_87A=Noun-Adj"). For the most basic word order features (81, 82 and 85–91), we additionally add a feature indicating if the order of the considered node pair agrees with the order specified by the WALS feature.[3]

While the training data for a standard preordering model consists of source sentences and their target-language order retrieved via word alignments, the training data for the universal reordering model is comprised of training examples from a large number of language pairs. Because of the diversity of this data, special care has to be taken to ensure a balanced dataset. We select an equal number of sentences from each language-specific training subcorpus. Additionally, we reduce class imbalance by further randomly shuffling the source tokens when creating training instances. This ensures a balanced distribution of classes in the training data. The distribution of the two classes is 84.5%/15.5% in the original and 50.1%/49.9% in the randomized dataset.

[2]Full set of features: words, word classes, dependency labels, POS tags, coarse POS tags, word and class of the left-most and right-most child token, and the tokens' distance to their parent.

[3]Example for WALS feature 87A=Adj-Noun: $f(a,b) = \begin{cases} \text{"W87A:ab"} & \text{if } a = \text{adj} \wedge b = \text{noun} \\ \text{"W87A:ba"} & \text{if } a = \text{noun} \wedge b = \text{adj} \end{cases}$



(a) Basic preordering model. (b) Universal reordering model.

Figure 9.2: Application of basic preordering and the universal reordering model.

9.4.3 Intrinsic Evaluation

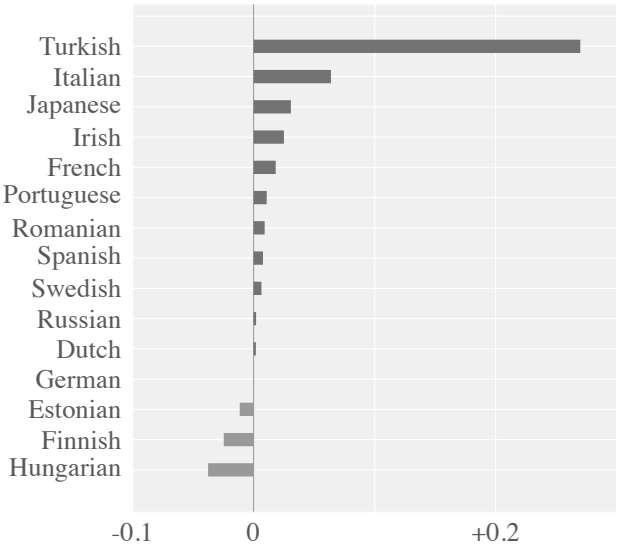
We use NPLM (Vaswani et al., 2013) to train a feed-forward neural network to predict the orientation of two nodes a and b based on the features described in Section 9.4.2. The network consists of 50 nodes on the input layer, 2 on the output layer, and 50 and 100 on the two hidden layers. We use a learning rate of 0.01, a batch size of 1000 and perform 60 training epochs, ensuring convergence of the log-likelihood on a validation set.

Preordering Data

The training data for the universal reordering model consists of a combined corpus of 30k sentence pairs each from the Tatoeba corpus (Tiedemann, 2012) for French, German, Japanese, Portuguese, Russian, and Spanish as well as 100k sentence pairs each from the OpenSubtitles 2012 corpus (Tiedemann, 2012) for Spanish, Portuguese, Italian, Danish, Romanian, Swedish, French, Greek, Russian, Polish, Arabic, Hebrew, Hungarian, Czech, Finnish, Icelandic, Dutch, Slovak, Chinese, German and Turkish. Word alignments for all corpora were produced using MGIZA++ (Och and Ney, 2003) using *grow-diag-final-and* symmetrization while performing 6, 6, 3 and 3 iterations of IBM Model 1, HMM, IBM Model 3 and IBM Model 4 respectively. To evaluate the model, we also use sets of manually word-aligned sentences for the following language pairs: English–Japanese (Neubig, 2011), English–German (Padó and Lapata, 2006), English–Italian (Farajian et al., 2014), English–French (Och and Ney, 2003), English–Spanish and English–Portuguese (Graça et al., 2008).

Quality of Word Order Predictions

Figure 9.3a shows the intrinsically measured quality of the predictions by the universal reordering model. We use Kendall τ (Kendall, 1938) to measure the correlation between the predicted word order and the *oracle* word order determined via the word alignments. Figure 9.3a plots absolute Kendall τ improvement over the original, i.e. unreordered, source sentence for the single best permutation for a number of language pairs. The three worst-performing target languages in Figure 9.3a, Estonian, Finnish and Hungarian, are all morphologically rich, indicating that additional considerations may be required to improve word order for languages of this type. Figure 9.3b shows



(a) Absolute 1-best Kendall τ improvements.

	Manual word alignments			Automatic word alignments		
Language	τ@10	τ@100	τ@1000	τ@10	τ@100	τ@1000
French	+01.95	+03.05	+03.05	+01.28	+02.12	+02.13
German	+04.03	+05.61	+05.61	+06.85	+08.27	+08.27
Italian	+06.39	+06.75	+06.75	+03.07	+03.32	+03.32
Portuguese	+05.87	+07.89	+07.89	+03.24	+03.55	+03.55
Spanish	+05.97	+06.57	+06.57	+04.28	+05.11	+05.11
Romanian	+02.49	+03.37	+03.37	+01.23	+02.09	+02.10
Swedish	+00.13	+00.42	+00.42	+00.71	+01.18	+01.18

(b) n-best permutation quality on manually and automatically aligned data.

Figure 9.3: Intrinsic word order quality (improvement over monotone permutation).

the quality of n-best permutations of the universal reordering model for both manually and automatically word-aligned sentence pairs. This table allows two observations: Firstly, the evaluation of word order quality using automatic alignments shows good agreement with the evaluation using manually word-aligned sentences, thus highlighting that automatic alignments should suffice for this purpose in most cases. Secondly, we can observe that for all datasets presented in this table little is gained from increasing the number of extracted permutations beyond 100 predictions. We therefore apply a maximum number of 100 permutations per sentence in all experiments presented in the rest of this chapter.

9.5 Translation Experiments

To evaluate the universal reordering model in a real-world task, we perform translation experiments on various language pairs. As a baseline system, we use a plain phrase-based machine translation system using a distortion-based reordering model with a distortion limit of 6. We compare against a baseline system without a lexicalized reordering model since the permutation lattices used in our system only delimit possible word order choices but do not score them. When applying the universal reordering model, we produce a lattice from each sentence's best 100 word order permutations. This lattice is then passed to the machine translation system and no additional reordering is allowed. During training, we choose the source sentence permutation closest to the gold word order determined via the word alignments (lattice silver training, see Chapter 6 or Daiber et al., 2016a). The word alignments for the preordered training corpus are then recreated from the original MGIZA++ alignments and the selected permutation.[4]

Translation experiments are performed with a phrase-based machine translation system, a version of Moses (Koehn et al., 2007) with extended lattice support.[5] We use the basic Moses features and perform 15 iterations of batch MIRA (Cherry and Foster, 2012). To control for optimizer instability, we perform 3 tuning runs for each system and report the mean BLEU score for these runs (Clark et al., 2011). As a baseline we use a translation system with distortion limit 6 and a distance-based reordering model. For each language pair, a 5-gram language model is estimated using *lmplz* (Heafield et al., 2013) on the target side of the parallel corpus.

9.5.1 Evaluating on a Broad Range of Languages

In order to test whether a single universal reordering model based on typological features can sufficiently generalize to multiple languages, we evaluate our model on a broad range of languages from various language families. While doing so, it is important to ensure that the results are not skewed by differences in the corpora used for training and testing each language pair. We therefore build translation systems from the same corpus and domain for every language pair.

We use the 2012 OpenSubtitles corpus[6] (Tiedemann, 2012) to extract 800k parallel sentences for each language pair, ensuring that every sentence pair contains only a single source sentence and that every source sentence contains at least 10 tokens. For each language pair, 10k parallel sentences are retained for tuning and testing. We use English as the source language in all language pairs. Table 9.2 summarizes properties of the data used in these experiments. Apart from the average source sentence length

[4] To keep the experiments manageable, we opted not to re-align the preordered training corpus using MGIZA++. Re-alignment often leads to improved translation results, therefore we are likely underestimating the potential preordered translation quality.

[5] Made available at `https://github.com/wilkeraziz/mosesdecoder`.

[6] `http://opus.lingfil.uu.se/OpenSubtitles2012.php`

	Sentences				Sentences		
Language	Total	Length	BiHDE	Language	Total	Length	BiHDE
Spanish	800k	14.29	0.57	Hebrew	800k	14.61	0.68
Portuguese	800k	14.29	0.58	Hungarian	800k	14.33	0.68
Italian	800k	14.68	0.61	Czech	800k	14.19	0.69
Danish	800k	14.50	0.62	Finnish	800k	14.36	0.69
Romanian	800k	14.24	0.64	Icelandic	800k	14.10	0.69
Swedish	800k	14.49	0.64	Dutch	800k	14.37	0.70
French	800k	14.25	0.65	Slovak	638k	15.08	0.70
Greek	800k	14.36	0.65	Chinese	636k	10.39	0.71
Russian	800k	15.08	0.65	German	800k	14.62	0.72
Polish	800k	14.22	0.67	Turkish	800k	14.25	0.72
Arabic	800k	14.84	0.68				

Table 9.2: Properties of training data (English source) from the 2012 OpenSubtitles corpus. We highlight the number of parallel sentence pairs, the average source sentence length and Bilingual Head Direction Entropy to indicate word order freedom.

and the number of training examples, we report Bilingual Head Direction Entropy (BiHDE, Daiber et al., 2016a), which indicates the difficulty of predicting a unique target word order given the source sentence and its syntactic analysis. The language pairs in Table 9.2 are sorted by their BiHDE score, meaning that target languages whose word order is more deterministic are listed first. For each language pair, we train four translation systems:

System: Baseline The baseline system is a standard phrase-based machine translation system with a distance-based reordering model, a distortion limit of 6, and a maximum phrase length of 7.

System: Gold The gold system provides an indication for the upper-bound achievable translation quality using preordering. In this system, the tuning and test sets are word-aligned along with the training portion of the corpus and the word alignments are then used to determine the optimal source word order. While this system provides an indication for the theoretically achievable improvement, this improvement may not be achievable in practice since not all information required to determine the target word order may be available on the source side (e.g. morphologically rich languages can allow several interchangeable word order variations). Apart from the source word order, the gold system is equivalent to the Baseline system.

System: No WALS As a baseline for our preordering systems, we create a translation system that differs from our universal reordering model only in the lack of WALS

	BLEU	Δ BLEU		
Language	Baseline	No WALS	WALS	Gold
Dutch	13.76	+0.11	+0.79	+3.44
Italian	23.59	+0.04	+0.48	+1.83
Turkish	5.89	−0.36	+0.43	+0.80
Spanish	23.82	−0.27	+0.29	+1.98
Portuguese	25.94	−0.48	+0.21	+1.64
Finnish	9.95	+0.13	+0.16	+0.51
Hebrew	11.64	+0.30	+0.11	+2.24
Romanian	16.11	+0.11	+0.11	+1.14
Hungarian	8.26	−0.10	+0.10	+0.61
Danish	26.36	−0.13	+0.08	+1.56
Chinese	11.09	−0.32	+0.05	+0.44
Greek	7.22	−0.02	+0.01	+0.49
Arabic	5.36	−0.10	−0.01	+0.36
Swedish	25.60	−0.14	−0.03	+2.04
Slovenian	10.56	−0.35	−0.10	+1.21
Slovak	15.56	−0.09	−0.13	+1.98
Icelandic	14.97	−0.31	−0.14	+0.66
Polish	17.68	−0.45	−0.16	+0.40
Russian	20.12	−0.47	−0.17	+0.92
German	17.08	−0.21	−0.19	+3.31
Czech	12.81	−0.47	−0.21	+0.70
French	19.92	−0.70	−0.23	+1.20

Table 9.3: Translation experiments with parallel subtitle corpora. The system labeled "WALS" is the universal reordering model with access to all WALS information for the target language, "No WALS" is a universally trained model without any typological information, "Gold" is a system trained with testing time target word order.

information. The preordering model is trained using the standard set of features described in Section 9.4.1 with only a single additional feature: the name of the target language. As in the WALS system, this system is applied by generating a minimized lattice from the 100-best permutations of each sentence and restricting the decoder's search space to this lattice. This system therefore isolates two potential sources of improvement: (1) improvement due to restricting the search space by the source dependency tree and (2) improvement from the preordering model itself, independent of the typology information provided by WALS.

System: WALS The WALS system applies the universal reordering model introduced in Section 9.4.2. For each language pair, the preordering model is provided with the target language and all the WALS features available for this language. The machine translation system's search space is then restricted using the minimized lattice of the 100-best word order permutations for each sentence and no additional reordering within the decoder is allowed.

The results of the translation experiments using the OpenSubtitles corpora are presented in Table 9.3. BLEU scores for the No WALS, WALS and Gold systems are reported as absolute improvement over the Baseline system (Δ BLEU). Over the three tuning runs performed for each model, we observe minor variance in BLEU scores between the runs (mean standard deviations: Baseline 0.04, No WALS 0.05, WALS 0.04, Gold 0.07), thus we report the mean BLEU score for each model's three runs.

While performing monotone decoding (i.e., allowing no reordering on top of the input lattice), the universal reordering model (WALS) enables improvements or comparable performance for the majority of the language pairs we evaluated while the No WALS system performs worse for most language pairs. This suggests that the improvements are not due to the neural preordering model or the lattice-based translation alone, but that the WALS information is crucial in enabling these results.

9.5.2 Influence of Domain and Data Size

While the experiments using the subtitle corpora presented in the previous section allow a fair comparison of a large number of language pairs, they also exhibit certain restrictions: (1) all experiments are limited to a single domain, (2) the source sentences are fairly short, and (3) to ensure consistent corpus sizes, a limited number of 800k sentence pairs had to be used. Therefore, we perform an additional set of experiments with data from different domains, longer sentences and a larger number of sentence pairs. To train the translation systems for these experiments, we use the following training data: For English–Italian, English–Spanish and English–Portuguese, we train systems on Europarl v7 (Koehn, 2005). English–Hungarian uses the WMT 2008 training data,[7] English–Turkish the SETIMES2 corpus (Tiedemann, 2009). Tuning is performed on

[7] http://www.statmt.org/

	Dataset				Baseline	WALS
Language	Domain	Size	Length	BiHDE	BLEU	Δ BLEU
Turkish	News	0.20m	23.54	0.73	8.27	+0.34
Spanish	Parl./news	1.73m	23.47	0.58	24.34	+0.18
Italian	Parl./news	1.67m	24.49	0.61	24.83	+0.13
Portuguese	Parl./news	1.73m	23.67	0.58	32.13	−0.08
Hungarian	Parl./news	1.41m	17.11	0.70	7.63	−0.19

Table 9.4: Translation experiments with varying training data sizes and domains (news and parliamentary proceedings).

the first 1512 sentences of newssyscomb2009+newstest2009 (English–Italian), newstest2009 (English–Spanish), newsdev2016 (English–Turkish), newstest2008 (English–Hungarian), and the first 3000 sentences of news commentary v11 (English–Portuguese). As test sets we use the rest of newssyscomb2009+newstest2009 (English–Italian), newstest2013 (English–Spanish), newstest2016 (English–Turkish), newstest2009 (English–Hungarian), and the first 3000 sentences of news commentary v11 not used in the dev set (English–Portuguese). All datasets are filtered to contain sentences up to 50 words long, and tokenization and truecasing is performed using the Moses tokenzier and truecaser. Statistics about each dataset and the dataset's domains, as well as translation results for the baseline system and the universal reordering model are summarized in Table 9.4. The results indicate that despite the longer sentences and different domains, the universal reordering model performs similarly as in the experiments performed in Section 9.5.1.

Our intrinsic evaluation (Section 9.4.3) as well as the extrinsic evaluation on a translation task (Section 9.5) indicate that a universal reordering model is not only feasible but can also provide good results on a diverse set of language pairs. The performance difference between the No WALS baseline and the universal reordering model (cf. Table 9.3) further demonstrates that the typological data points provided by WALS are the crucial ingredient in enabling this model to work.

9.6 Conclusion

In this chapter, we have shown that linguistics in the form of linguistic typology and modern methods in natural language processing in the form of neural networks are not rivaling approaches but can come together in a symbiotic manner. In the best case, combining both approaches can yield the best of both worlds: the generalization power of linguistic descriptions and the good empirical performance of statistical models. Concretely, we have shown in this chapter that it is possible to use linguistic typology information as input to a preordering model, thus enabling us to build a single model with a single set of model parameters for a diverse range of languages. As an empir-

ical result, our findings provide support for the adequacy of the language descriptions found in linguistic typology. Additionally, they open the way for more compact and universal models of word order that can be especially beneficial for machine translation between language pairs with little parallel data. Finally, our results also suggest that target-language typological features could play a key role in building better preordering models.

Chapter 10

Conclusion

This thesis focuses on statistical machine translation. It contributes an extensive study of the impact of typological factors in two main areas: differences in word order freedom and morphological complexity. In both areas, we demonstrate that typological differences between languages influence translation quality and show that these differences can and should be taken into account in order to produce more typologically robust translation models.

We study the following typological differences and present methods to address them:

- In the area of word order freedom, we show that producing a space of potential word order choices instead of a single word order can improve how well preordering generalizes to typologically diverse target languages. We further show that word order permutation lattices provide a suitable representation for efficiently integrating such word order choices into a phrase-based machine translation system. Using translation experiments and with a newly introduced entropy measure for word order freedom, we show that these word order permutation lattices are effective for both strict and free word order target languages.
- In the area of inflectional word formation processes and specifically for translation into morphologically rich languages, we show that morphologically impoverished source languages can be enriched with unexpressed morphological attributes to mitigate the translation model's difficulties in selecting appropriate word forms. The set of morphological attributes whose addition to the source language may potentially be helpful in this task varies from language pair to language pair. Hence, we propose a latent variable model for selecting these morphological attributes from parallel data, which results in a high-quality selection in our experiments.

- In the area of non-inflectional word formation processes, which can cause further sparsity issues for phrase-based machine translation systems, we introduce an unsupervised approach to split compound words into their individual parts using distributional semantics. We show that semantic analogies ("bookshop is to shop as bookshelf is to shelf"), which can be performed using word embeddings obtained from large monolingual corpora, provide a rich source of semantic information for surfacing the internal structure of compound words. The resulting semantic information is sufficient to reliably isolate the parts of compounds and thus enables translation systems to work with comparable translation units in the source and target language.

The methods we introduce in this thesis serve to build machine translation models that are more robust to typological differences between languages. Various linguistic theories and the empirical findings of linguistic typology suggest that the similarities and differences between languages can in many cases be captured with a small set of parameters. If this is indeed the case, then models for natural language processing should not only be able to reliably deal with typologically diverse languages, but they should also be able to benefit from the existence of such a small set of parameters and the knowledge about their values which was collected in linguistic typology. We show that for word order, specifically for preordering in phrase-based models, typological information collected by linguists in the World Atlas of Language Structures, when combined with neural network techniques, can be used to build universal reordering models that perform well on a typologically diverse set of target languages. Thus, in this instance typological knowledge can be used not only to build models that work for more language pairs but enables models that can use this information to improve generalization and that require less training data.

This thesis furthermore provides a comprehensive summary of the relationship between linguistic typology and machine translation. While focusing on phrase-based statistical machine translation, it introduces the preliminaries of three major paradigms in machine translation and highlights the structure each of these approaches imposes on the translation process. It examines how linguistic structure in particular is treated in various approaches and covers how the various areas studied in linguistic typology relate to machine translation and which problems they may cause. Finally, we have provided a broad overview of how the issues caused by significant typological differences are addressed in machine translation and how our work fits into the larger context of machine translation research.

Bibliography

Roee Aharoni and Yoav Goldberg. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-2021.

Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1972.

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer, 2007. http://www.openfst.org.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. Many languages, one parser. *Transactions of the Association of Computational Linguistics*, 4:431–444, 2016. URL http://aclanthology.coli.uni-saarland.de/pdf/Q/Q16/Q16-1031.pdf.

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. Zero-resource dependency parsing: Boosting delexicalized cross-lingual transfer with linguistic knowledge. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 119–130. The COLING 2016 Organizing Committee, 2016. URL http://aclanthology.coli.uni-saarland.de/pdf/C/C16/C16-1012.pdf.

Eleftherios Avramidis and Philipp Koehn. Enriching morphologically poor languages for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 763–770, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P08-1087.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Proceedings of the International Conference on Learning Representations*, 2015.

Egon Balas and Paolo Toth. Branch and bound methods for the traveling salesman problem. Technical report, Carnegie-Mellon Univ. Pittsburgh PA Management Sciences Research Group., 1983.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-2322.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W05/W05-0909.

Yehoshua Bar-Hillel, M. Perles, and E. Shamir. On formal properties of simple phrase structure grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14:143–172, 1961. Reprinted in Y. Bar-Hillel. (1964). *Language and Information: Selected Essays on their Theory and Application*, Addison-Wesley 1964, 116–150.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1947–1957, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1208.

Leonard E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In Oved Shisha, editor, *Inequalities III: Proceedings of the Third Symposium on Inequalities*, pages 1–8, University of California, Los Angeles, 1972. Academic Press.

Emily M. Bender. Linguistically naïve != language independent: Why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 26–32, Athens, Greece, March 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W09-0106.

Emily M. Bender. Linguistic typology in natural language processing. *Linguistic Typology*, 20(3):645–660, 2016.

Emily M. Bender, Daniel P. Flickinger, and Stephan Oepen. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan, 2002.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944966.

Nicola Bertoldi, Richard Zens, and Marcello Federico. Speech translation by confusion network decoding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4 of *ICASSP '07*, pages 1297–1300, Honolulu, HI, April 2007. IEEE.

Alexandra Birch, Miles Osborne, and Phil Blunsom. Metrics for MT evaluation: Evaluating reordering. *Machine Translation*, 24(1):15–26, March 2010. ISSN 0922-6567. doi: 10.1007/s10590-009-9066-5. URL http://dx.doi.org/10.1007/s10590-009-9066-5.

Ondřej Bojar and Kamil Kos. 2010 failures in English-Czech phrase-based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 60–66, Stroudsburg, PA, USA, 2010. ISBN 978-1-932432-71-8. URL http://dl.acm.org/citation.cfm?id=1868850.1868855.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation*, pages 199–231, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2302.

Dunstan Brown. Morphological typology. In Jae Jung Song, editor, *The Oxford Handbook of Linguistic Typology*. Oxford University Press, 2010. ISBN 9780199281251. URL http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199281251.001.0001/oxfordhb-9780199281251-e-023.

Peter F. Brown, John Cocke, Stephen A. Della-Pietra, Vincent J. Della-Pietra, Frederick Jelinek, Robert L. Mercer, and Paul Rossin. A statistical approach to language translation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 1988.

Peter F. Brown, John Cocke, Stephen A. Della-Pietra, Vincent J. Della-Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul Rossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):76–85, 1990.

Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

Peter F. Brown, Stephen A. Della-Pietra, Vincent J. Della-Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–313, 1993. URL http://www.aclweb.org/anthology/J93-2003.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of BLEU in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy, April 2006. Association for Computational Linguistics. URL http://aclweb.org/anthology-new/E/E06/E06-1032.

Fabienne Cap, Alexander Fraser, Marion Weller, and Aoife Cahill. 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*, pages 579–587, Gothenburg, Sweden, April 2014a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E14-1061.

Fabienne Cap, Marion Weller, Anita Ramm, and Alexander Fraser. CimS – the CIS and IMS joint submission to WMT 2014 translating from English into German. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 71–78, Baltimore, Maryland, USA, June 2014b. URL http://www.aclweb.org/anthology/W/W14/W14-3305.

Marine Carpuat and Dekai Wu. Context-dependent phrasal translation lexicons for statistical machine translation. *Proceedings of Machine Translation Summit XI*, pages 73–80, 2007.

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D Manning. Phrasal: A statistical machine translation toolkit for exploring new model. *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 9–12, 2010.

Victor Chahuneau, Eva Schlinger, Noah A. Smith, and Chris Dyer. Translating into morphologically rich languages with synthetic phrases. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1677–1687, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1174.

Colin Cherry and George Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N12-1047.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics, 2005.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33 (2):201–228, 2007. ISSN 0891-2017. doi: 10.1162/coli.2007.33.2.201. URL http://dx.doi.org/10.1162/coli.2007.33.2.201.

Noam Chomsky. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, MA, 1965.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-2031.

John Cocke and Jacob T. Schwartz. Programming languages and their compilers: Preliminary notes. Technical report, Courant Institute of Mathematical Sciences, New York University, 1970.

Michael Collins, Philipp Koehn, and Ivona Kucerova. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P05-1066.

Bernard Comrie. *Language Universals and Linguistic Typology: Syntax and Morphology*. Blackwell, Oxford, 1981. ISBN 0631129715 063112618.

Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

Marta R. Costa-jussà and José A. R. Fonollosa. Statistical machine reordering. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 70–76, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W06/W06-1609.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7: 551–585, 2006.

Sonia Cristofaro. Language universals and linguistic knowledge. In Jae Jung Song, editor, *The Oxford Handbook of Linguistic Typology*. Oxford University Press, 2010. ISBN 9780199281251. URL http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199281251.001.0001/oxfordhb-9780199281251-e-013.

Joachim Daiber and Khalil Sima'an. Delimiting morphosyntactic search space with source-side reordering models. In Jan Hajič and António Branco, editors, *Proceedings of the 1st Deep Machine Translation Workshop*, pages 29–38, Praha, Czechia, 2015a. ÚFAL MFF UK, ÚFAL MFF UK. ISBN 978-80-904571-7-1. URL http://www.aclweb.org/anthology/W15-5704.

Joachim Daiber and Khalil Sima'an. Machine translation with source-predicted target morphology. In *Proceedings of the 15th Machine Translation Summit (MT Summit 2015)*, pages 283–296, Miami, USA, 2015b.

Joachim Daiber and Rob van der Goot. The denoised web treebank: Evaluating dependency parsing under noisy input conditions. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. Splitting compounds by semantic analogy. In Jan Hajič and António Branco, editors, *Proceedings of the 1st Deep Machine Translation Workshop*, pages 20–28, Praha, Czechia, 2015. ÚFAL MFF UK, ÚFAL MFF UK. ISBN 978-80-904571-7-1. URL http://www.aclweb.org/anthology/W15-5703.

Joachim Daiber, Miloš Stanojević, Wilker Aziz, and Khalil Sima'an. Examining the relationship between preordering and word order freedom in machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 118–130, Berlin, Germany, August 2016a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-2213.

Joachim Daiber, Miloš Stanojević, and Khalil Sima'an. Universal reordering via linguistic typology. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3167–3176, Osaka, Japan, December 2016b. The COLING 2016 Organizing Committee. URL http://aclweb.org/anthology/C16-1298.

Michael Daniel. Linguistic typology and the study of language. In Jae Jung Song, editor, *The Oxford Handbook of Linguistic Typology*. Oxford University Press, 2010. ISBN 9780199281251. URL http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199281251.001.0001/oxfordhb-9780199281251-e-004.

Hal Daume III and Lyle Campbell. A bayesian model for discovering typological implications. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 65–72, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P07-1009.

Adrià De Gispert, Gonzalo Iglesias, and Bill Byrne. Fast and accurate preordering for SMT using neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1012–1017, Denver, Colorado, May–June 2015. URL http://www.aclweb.org/anthology/N15-1105.

Simon DeDeo, Robert X. D. Hawkins, Sara Klingenstein, and Tim Hitchcock. Bootstrap methods for the empirical study of decision-making and information flows in social systems. *Entropy*, 15(6):2246–2276, 2013. ISSN 1099-4300. doi: 10.3390/e15062246. URL http://www.mdpi.com/1099-4300/15/6/2246.

John DeNero and Jakob Uszkoreit. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1018.

John DeNero, David Chiang, and Kevin Knight. Fast consensus decoding over translation forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 567–575, Stroudsburg, PA, USA, 2009. ISBN 978-1-932432-46-6. URL http://dl.acm.org/citation.cfm?id=1690219.1690226.

Michael Denkowski and Alon Lavie. METEOR 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2107.

Michael Denkowski and Alon Lavie. METEOR universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-3348.

Bonnie J. Dorr, Pamela W. Jordan, and John W Benoit. A survey of current paradigms in machine translation. *Advances in computers*, 49:1–68, 1999.

Wolfgang U. Dressler, Laura E. Lettner, and Katharina Korecky-Kröll. Acquisition of german diminutive formation and compounding in a comparative perspective. Evidence for typology and the role of frequency. In Ferenc Kiefer, Mária Ladányi, and Péter Siptár, editors, *Current Issues in Morphological Theory. (Ir)Regularity, analogy and frequency. Selected papers from the 14th International Morphology Meeting, Budapest, 13-16 May 2010*, pages 237–264. Benjamins, Amsterdam, 2012.

Matthew S. Dryer. The Greenbergian word order correlations. *Language*, 68(1):81–183, 1992.

Matthew S. Dryer and Martin Haspelmath, editors. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL `http://wals.info/`.

Ondřej Dušek, Zdeněk Žabokrtský, Martin Popel, Martin Majliš, Michal Novák, and David Mareček. Formemes in english-czech deep syntactic mt. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 267–274, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2393015.2393052`.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N13-1073`.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio, June 2008.

Christopher J. Dyer. The "noisier channel": Translation from morphologically complex languages. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 207–211, Prague, Czech Republic, June 2007.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1078`.

M. Amin Farajian, Nicola Bertoldi, and Marcello Federico. Online word alignment for online adaptive machine translation. In *Proceedings of the EACL 2014 Workshop*

on Humans and Computer-assisted Translation, pages 84–92, Gothenburg, Sweden, April 2014. URL http://www.aclweb.org/anthology/W14-0313.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. IRSTLM: An open source toolkit for handling large scale language models. In *Proceedings of Interspeech 2008 - 9th Annual Conference of the International Speech Communication Association*, 2008.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1101.

Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. Modeling inflection and word-formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674, Avignon, France, April 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E12-1068.

Fabienne Fritzinger and Alexander Fraser. How to avoid burning ducks: Combining linguistic analysis and corpus statistics for German compound processing. In *Proceedings of the ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics (MATR)*, 2010.

Richard Futrell, Kyle Mahowald, and Edward Gibson. Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala, Sweden, August 2015. Uppsala University, Uppsala, Sweden. URL http://www.aclweb.org/anthology/W15-2112.

Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D08-1089.

Michel Galley and Christopher D. Manning. Accurate non-hierarchical phrase-based translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974, Los Angeles, California, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N10-1140.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic

translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P06/P06-1121.

Qin Gao and Stephan Vogel. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics, 2008.

Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 123–135, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1012.

João Graça, Joana Paulo Pardal, and Luísa Coheur. Building a golden collection of parallel multi-language word alignments, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.310.1300.

Giorgio Graffi. The pioneers of linguistic typology: From Gabelentz to Greenberg. In Jae Jung Song, editor, *The Oxford Handbook of Linguistic Typology*. Oxford University Press, 2010. ISBN 9780199281251. URL http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199281251.001.0001/oxfordhb-9780199281251-e-003.

Joseph H. Greenberg. Some universals of grammar with particular reference to the order of meaningful elements. In *Universals of Human Language*, pages 73–113. MIT Press, Cambridge, MA, 1963.

Joseph H. Greenberg. *Universals of Language*. The MIT Press, Cambridge, MA, 2nd edition, 1966a.

Joseph H. Greenberg. *Language Universals: with Special Reference to Feature Hierarchies*. Mouton de Gruyter, Den Haag, 1966b.

Jan Hajič. Tectogrammatical representation: towards a minimal transfer in machine translation. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6)*, pages 216–226, Universitá di Venezia, May 2002. URL http://www.aclweb.org/anthology/W02-2231.

J.A. Hawkins. *Word Order Universals*. Quantitative analyses of linguistic structure. Academic Press, 1983. ISBN 9780123333704. URL https://books.google.nl/books?id=-lhiAAAAMAAJ.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the*

51st Annual Meeting of the Association for Computational Linguistics, pages 690–696, Sofia, Bulgaria, August 2013. URL http://kheafield.com/professional/edinburgh/estimate_paper.pdf.

Verena Henrich and Erhard W. Hinrichs. Determining immediate constituents of compounds in GermaNet. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing 2011*, 2011.

Teresa Herrmann, Jochen Weiner, Jan Niehues, and Alex Waibel. Analyzing the potential of source sentence reordering in statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

Mark Hopkins and Jonathan May. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1125.

Bryant Huang and Kevin Knight. Relabeling syntax trees to improve syntax-based machine translation quality. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 240–247, New York City, USA, June 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N06/N06-1031.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325, 2005.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. Head finalization: A simple reordering rule for sov languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-1736.

Roman Jakobson. *Kindersprache, Aphasie und allgemeine Lautgesetze*. Almqvist & Wiksell, Uppsala, 1941.

Laura Jehl, Adrià de Gispert, Mark Hopkins, and Bill Byrne. Source-side preordering for translation using logistic regression and depth-first branch-and-bound search. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 239–248, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E14-1026.

Minwoo Jeong, Kristina Toutanova, Hisami Suzuki, and Chris Quirk. A discriminative lexicon model for complex morphology. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, 2010.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. Technical report, Google, 2016. URL https://arxiv.org/abs/1611.04558.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009. ISBN 0131873210.

Tadao Kasami. An efficient recognition and syntax-analysis algorithm for context-free languages. Technical report, Air Force Cambridge Research Lab, Bedford, MA, 1965.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

Maxim Khalilov and Khalil Sima'an. Statistical translation after source reordering: Oracles, context-aware models, and empirical analysis. *Natural Language Engineering*, 18:491–519, 10 2012. ISSN 1469-8110. doi: 10.1017/S1351324912000162. URL http://journals.cambridge.org/article_S1351324912000162.

Maxim Khalilov, José A. R. Fonollosa, and Mark Dras. Coupling hierarchical word reordering and decoding in phrase-based statistical machine translation. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, SSST '09, pages 78–86, Stroudsburg, PA, USA, 2009. ISBN 978-1-932432-39-8. URL http://dl.acm.org/citation.cfm?id=1626344.1626354.

Paul V. Kiparsky. The rise of positional licensing. In Ans van Kemenade and Nigel Vincent, editors, *Parameters of Morphosyntactic Change*. Oxford University Press, Oxford, 1997.

Kevin Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615, December 1999. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=973226.973232.

Kevin Knight and Yaser Al-Onaizan. Translation with finite-state devices. In *Proceedings of the Association for Machine Translation in the Americas*, AMTA, pages 421–437, Langhorne, PA, USA, 1998.

Philipp Koehn. *Noun Phrase Translation*. PhD thesis, Los Angeles, CA, USA, 2003. AAI3133297.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395. Association for Computational Linguistics, 2004a.

Philipp Koehn. Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *AMTA*, pages 115–124, 2004b.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, Cambridge, 2010.

Philipp Koehn and Kevin Knight. Empirical methods for compound splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2003.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA, 2003.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. URL `http://dl.acm.org/citation.cfm?id=1557769.1557821`.

Vladislav Kuboň and Markéta Lopatková. Free or fixed word order: What can treebanks reveal? In Jakub Yaghob, editor, *ITAT 2015: Information Technologies – Applications and Theory, Proceedings of the 15th conference ITAT 2015*, volume 1422 of *CEUR Workshop Proceedings*, pages 23–29, Praha, Czechia, 2015. Charles University in Prague, CreateSpace Independent Publishing Platform. ISBN 978-1515120650.

Shankar Kumar and William Byrne. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 63–70, Stroudsburg, PA, USA, 2003.

Gorka Labaka, Oneka Jauregi, Arantza Díaz de Ilarraza, Michael Ustaszewski, Nora Aranberri, and Eneko Agirre. Deep-syntax tectomt for english-spanish MT. In Jan Hajič and António Branco, editors, *Proceedings of the 1st Deep Machine Translation Workshop*, pages 55–63, Praha, Czechia, 2015. ÚFAL MFF UK, ÚFAL MFF UK. ISBN 978-80-904571-7-1. URL `http://www.aclweb.org/anthology/W15-5707`.

K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

Winfred P. Lehmann. A structural principle of language and its implications. *Language*, 49:47–66, 1973.

Winfred P. Lehmann. *Syntactic typology: studies in the phenomenology of language*. University of Texas Press, 1978. ISBN 9780292775459.

Uri Lerner and Slav Petrov. Source-side classifier preordering for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 513–523, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1049.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220271. URL http://www.aclweb.org/anthology/P06-1096.

Rochelle Lieber and Pavol Štekauer. *The Oxford Handbook of Compounding*. Oxford University Press, 2009.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, March 2002. ISSN 1532-4435. doi: 10.1162/153244302760200687. URL http://dx.doi.org/10.1162/153244302760200687.

Adam Lopez. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 976–985, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D07/D07-1104.

Adam Lopez. Translation as weighted deduction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 532–540, Athens, Greece, March 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E09-1061.

Xia Lu. Exploring word order universals: a probabilistic graphical model approach. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 150–157, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-3022.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1166.

Matous Machacek and Ondrej Bojar. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, Maryland, USA, June 2014. URL http://www.aclweb.org/anthology/W/W14/W14-3336.

David Mareček. Improving word alignment using alignment of deep structures. In *Proceedings of the 12th International Conference, TSD 2009*, pages 56–63, 2009.

David Mareček, Martin Popel, and Zdeněk Žabokrtský. Maximum entropy translation model in dependency-based mt framework. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 201–201, Uppsala, Sweden, 2010. Association for Computational Linguistics. ISBN 978-1-932432-71-8.

José B Marino, Rafael E Banchs, Josep M Crego, Adria de Gispert, Patrik Lambert, José AR Fonollosa, and Marta R Costa-Jussà. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549, 2006.

Andre Martins, Noah Smith, and Eric Xing. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 342–350, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P09/P09-1039.

Andre Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mario Figueiredo. Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 34–44, Cambridge, MA, October 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D10-1004.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 75–82, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P05-1010.

Ryan McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK., July

2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1006.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013.

Mehryar Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, January 2002.

Edith A. Moravcsik. Explaining language universals. In Jae Jung Song, editor, *The Oxford Handbook of Linguistic Typology*. Oxford University Press, 2010. ISBN 9780199281251. URL http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199281251.001.0001/oxfordhb-9780199281251-e-005.

Stefan Müller. *Grammatical Theory: From Transformational Grammar to Constraint-Based Approaches*. Number 1 in Lecture Notes in Language Sciences. Language Science Press, Berlin, 2015. URL http://hpsg.fu-berlin.de/~stefan/Pub/grammatical-theory.html. Open Review Version.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1032.

Tahira Naseem, Regina Barzilay, and Amir Globerson. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629–637, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P12-1066.

Graham Neubig. The Kyoto free translation task. http://www.phontron.com/kftt, 2011.

Graham Neubig and Kevin Duh. On the elements of an accurate tree-to-string machine translation system. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–149, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-2024.

Graham Neubig and Taro Watanabe. Optimization for statistical machine translation: A survey. *Computational Linguistics*, 42(1):1–54, March 2016. ISSN 0891-2017. doi: 10.1162/COLI_a_00241. URL http://dx.doi.org/10.1162/COLI_a_00241.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-2093.

Hermann Ney. Speech translation: coupling of recognition and translation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520, Phoenix, AZ, March 1999. IEEE.

Sonja Nießen and Hermann Ney. Improving SMT quality with morpho-syntactic analysis. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, 2000.

Joakim Nivre and Jens Nilsson. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 99–106, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P05-1013.

Franz Josef Och. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. PhD thesis, RWTH Aachen Department of Computer Science, Aachen, Germany, 2002. URL http://www-i6.informatik.rwth-aachen.de/web/PhD/phd_theses.html.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075117. URL http://www.aclweb.org/anthology/P03-1021.

Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 295–302, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073133. URL http://dx.doi.org/10.3115/1073083.1073133.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4), 2004. URL http://aclanthology.coli.uni-saarland.de/pdf/J/J04/J04-4002.pdf.

Helen O'Horan, Yevgeni Berzak, Ivan Vulic, Roi Reichart, and Anna Korhonen. Survey on the use of typological information in natural language processing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1297–1308, Osaka, Japan, December 2016. The

COLING 2016 Organizing Committee. URL http://aclweb.org/anthology/C16-1123.

Robert Östling. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China, July 2015. URL http://www.aclweb.org/anthology/P15-2034.

Sebastian Padó and Mirella Lapata. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1161–1168, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P06-1146.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002.

Karl Pearson. Contributions to the mathematical theory of evolution—III, regression, heredity and panmixia. *Philosophical Transactions of the Royal Society of London*, 187:253–318, 1896.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P06-1055.

Martin Popel and Zdeněk Žabokrtský. Tectomt: Modular nlp framework. In *Proceedings of the 7th International Conference on Advances in Natural Language Processing*, IceTAL'10, pages 293–304, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-14769-0, 978-3-642-14769-2. URL http://dl.acm.org/citation.cfm?id=1884371.1884406.

Maja Popović, Daniel Stein, and Hermann Ney. Statistical machine translation of German compound words. In *Proceedings of FinTal - 5th International Conference on Natural Language Processing*, 2006.

Detlef Prescher. Inducing head-driven pcfgs with latent heads: Refining a tree-bank grammar for parsing. In *In ECML'05*, 2005.

Chris Quirk, Arul Menezes, and Colin Cherry. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of*

the Association for Computational Linguistics (ACL'05), pages 271–279, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219874. URL http://www.aclweb.org/anthology/P05-1034.

Taraka Rama and Prasanth Kolachina. How good are typological distances for determining genealogical relationships among languages? In *Proceedings of COLING 2012: Posters*, pages 975–984, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-2095.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. An empirical study on compositionality in compound nouns. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, 2011.

Edward Sapir. *Language: An Introduction to the Study of Speech*. Harvest books. Harcourt, Brace and Co, New York City, USA, 1921.

Sabine Schulte im Walde, Stefan Müller, and Stephen Roller. Exploring vector space models to predict the compositionality of German noun-noun compounds. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM)*, 2013.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681, 1997.

Rico Sennrich and Barry Haddow. A joint dependency model of morphological and syntactic structure for statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2087, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1248.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1162.

Libin Shen, Jinxi Xu, and Ralph Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-1066.

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. Translating with

non-contiguous phrases. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005. URL http://aclweb.org/anthology/H05-1095.

Klaus Simon. An improved algorithm for transitive closure on acyclic digraphs. *Theoretical Computer Science*, 58(1-3):325–346, June 1988. ISSN 0304-3975. doi: 10.1016/0304-3975(88)90032-1. URL http://dx.doi.org/10.1016/0304-3975(88)90032-1.

Adam Smith. *The theory of moral sentiments [microform] : to which is added a dissertation on the origin of languages / by Adam Smith*. Printed for A. Millar, A. Kincaid and J. Bell, and sold by T. Cadell London, 3rd ed. edition, 1767.

Jason R. Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-1135.

Benjamin Snyder. *Unsupervised Multilingual Learning*. PhD thesis, Electrical Engineering and Computer Science, MIT, 2010.

Anders Søgaard and Julie Wulff. An empirical etudy of non-lexical extensions to delexicalized transfer. In *Proceedings of COLING 2012: Posters*, pages 1181–1190, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-2115.

Jae Jung Song. Word order typology. In Jae Jung Song, editor, *The Oxford Handbook of Linguistic Typology*. Oxford University Press, 2010. ISBN 9780199281251. URL http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199281251.001.0001/oxfordhb-9780199281251-e-014.

Radu Soricut and Franz Och. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 299–305, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/P16-2049.

Miloš Stanojević and Khalil Sima'an. BEER: BEtter Evaluation as Ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419,

Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-3354.

Miloš Stanojević and Khalil Sima'an. Reordering grammar induction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 44–54, Lisbon, Portugal, September 2015. URL http://aclweb.org/anthology/D15-1005.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N13-1126.

Yee Whye Teh, Hal Daumé III, and Daniel Roy. Bayesian agglomerative clustering with coalescents. In *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, 2007.

Jörg Tiedemann. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, 2009. ISBN 978 90 272 4825 1.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7. URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

Christoph Tillman. A unigram orientation model for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

Christoph Tillmann and Hermann Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29:97–133, 2003.

Christoph Tillmann, Stephan Vogel, Hermann Ney, and Alex Zubiaga. A DP-based search using monotone alignments in statistical translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1997. URL http://acl.ldc.upenn.edu/P/P97/P97-1037.pdf.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P08-1059.

Ke Tran, Arianna Bisazza, and Christof Monz. Word translation prediction for morphologically rich languages with bilingual neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014.

Roy Tromble and Jason Eisner. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D09-1105.

Rob van der Goot and Gertjan van Noord. Parser adaptation for social media by integrating normalization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 491–497, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-2078.

Edo S. van der Poort, Marek Libura, Gerard Sierksma, and Jack A. A. van der Veen. Solving the k-best traveling salesman problem. *Computers & Operations Research*, 26(4):409 – 425, 1999. ISSN 0305-0548. doi: http://dx.doi.org/10.1016/S0305-0548(98)00070-7. URL http://www.sciencedirect.com/science/article/pii/S0305054898000707.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, Washington, USA, October 2013. URL http://www.aclweb.org/anthology/D13-1140.

Bernard Vauquois. A survey of formal grammars and algorithms for recognition and transformation in mechanical translation. In *IFIP Congress (2)*, pages 1114–1122, 1968.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics, 1996.

Georg von der Gabelentz. Hypologie der Sprachen, eine neue Aufgabe der Linguistik. *Indogermanische Forschungen*, 4:1–7, 1894.

Zdeněk Žabokrtský and Martin Popel. Hidden markov tree model in dependency-based machine translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 145–148, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P09/P09-2037.

Zdeněk Žabokrtský, Jan Ptacek, and Petr Pajas. TectoMT: Highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W08/W08-0325.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation*, pages 505–510, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-2342.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D07/D07-1080.

Warren Weaver. *Translation (1949). Reproduced in W.N. Locke, A.D. Booth (eds.)*. MIT Press, 1955.

Marion Weller, Fabienne Cap, Stefan Müller, Sabine Schulte im Walde, and Alexander Fraser. Distinguishing degrees of compositionality in compound splitting for statistical machine translation. In *Proceedings of the First Workshop on Computational Approaches to Compound Analysis (ComaComa) at COLING*, 2014.

Philip Williams and Philipp Koehn. Agreement constraints for statistical machine translation into german. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 217–226, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2126.

Philip Williams, Rico Sennrich, Matt Post, and Philipp Koehn. *Syntax-based Statistical Machine Translation*, volume 9 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, 4 edition, 8 2016. doi: 10.2200/S00716ED1V04Y201604HLT033.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL `http://arxiv.org/abs/1609.08144`.

Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1220355.1220428. URL `http://dx.doi.org/10.3115/1220355.1220428`.

Jia Xu, Evgeny Matusov, Richard Zens, and Hermann Ney. Integrated chinese word segmentation in statistical machine translation. In *International Workshop on Spoken Language Translation*, Pittsburgh, 2005.

David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *HLT '01: Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1072133.1072187. URL `http://dx.doi.org/10.3115/1072133.1072187`.

Reyyan Yeniterzi and Kemal Oflazer. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 454–464, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P10-1047`.

Daniel H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208, 1967.

Mikhail Zaslavskiy, Marc Dymetman, and Nicola Cancedda. Phrase-based statistical machine translation as a traveling salesman problem. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 333–341, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P09/P09-1038`.

Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, 2008.

Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *IJCNLP 2008 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India, 2008. International Institute of Information Technology.

Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. HamleDT: To parse or not to parse? In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Richard Zens and Hermann Ney. A comparative study on reordering constraints in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 144–151, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075115. URL http://www.aclweb.org/anthology/P03-1019.

Hao Zhang and Daniel Gildea. Factorization of synchronous context-free grammars in linear time. In *NAACL Workshop on Syntax and Structure in Statistical Translation (SSST)*, pages 25–32, 2007. URL http://www.cs.rochester.edu/~gildea/pubs/zhang-gildea-ssst07.pdf.

Yuqi Zhang, Richard Zens, and Hermann Ney. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, SSST '07, pages 1–8, Stroudsburg, PA, USA, 2007. URL http://dl.acm.org/citation.cfm?id=1626281.1626282.

Andreas Zollmann and Ashish Venugopal. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W06/W06-3119.

Andreas Zollmann, Ashish Venugopal, Stephan Vogel, and Alex Waibel. The CMU-UKA syntax augmented machine translation system for the IWSLT-06. In *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, November 2006. URL http://20.210-193-52.unknown.qala.com.sg/archive/iwslt_06/papers/slt6_138.pdf.

Daniel Zwillinger and Stephen Kokoska. *CRC Standard Probability and Statistics Tables and Formulae*. CRC Press, 1999. ISBN 9781420050264.

Abstract

Machine translation systems often incorporate modeling assumptions motivated by properties of the language pairs they initially target. When such systems are applied to language families with considerably different properties, translation quality can deteriorate. Phrase-based machine translation systems, for instance, are ill-equipped to handle the challenges caused by relaxed word order constraints and productive word formation processes in morphologically rich languages. In this thesis, we ask what role the properties of languages, as studied in the field of linguistic typology, play in how well machine translation systems perform. We focus in particular on word order and morphology, and show that typological differences in these areas can be bridged by making certain linguistic phenomena overt to the translation system. Understanding and exploiting typological differences between languages enables improvements to the typological robustness of translation systems without significantly changing the assumptions of the underlying translation models.

We begin by studying the effect of word order freedom on preordering, a popular technique to model word order in phrase-based machine translation. We show that producing a space of potential word order choices instead of a single word order and integrating this space into the translation model via word order permutation lattices provides a principled way of improving the typological robustness of preordering.

Then, we show that reducing the dissimilarity between the source and target language in the area of morphological complexity improves phrase-based machine translation for typologically diverse language pairs. For inflectional morphology, we do so by enriching the morphologically impoverished source language with unexpressed morphological attributes, which enables better lexical choice in the target language. For non-inflectional morphology, we introduce a semantically motivated model of compounding, which can be used to split compound words into their meaning-carrying subparts, thus enabling the translation system to work with comparable translation units in the source and target language.

Finally, we show that besides helping to bridge the performance gaps between typologically diverse languages, linguistic typology can also serve as a source of knowledge

to guide reordering models and to facilitate universal reordering models applicable to multiple target languages. Such universal reordering models can learn in a data-driven manner which aspects of linguistic typology to pay attention to, enable better generalization and require less training data than models for individual languages.

Samenvatting

Typologisch robuuste statistische machinevertaling

Variaties en overeenkomsten tussen talen begrijpen en benutten voor machinevertaling

Machinevertaalsystemen gebruiken vaak modelleringsaannames die gebaseerd zijn op de taalparen waar ze oorspronkelijk voor gemaakt zijn. Als zulke systemen toegepast worden op taalfamilies met aanzienlijk verschillende eigenschappen, kan dat nadelig zijn voor de kwaliteit van de vertaling. Phrase-based machinevertaalsystemen zijn bijvoorbeeld slecht toegerust voor de uitdagingen die meegebracht worden door versoepelde woordvolgorderestricties en productieve woordvormingsprocessen in morfologisch rijke talen. In deze dissertatie vragen we welke rol taaleigenschappen, zoals bestudeerd in het veld van taaltypologie, in de prestaties van machinevertaalsystemen spelen. We leggen de nadruk op woordvolgorde en morfologie in het bijzonder en we laten zien dat typologische verschillen in deze gebieden overbrugd kunnen worden door bepaalde taalverschijnselen expliciet te maken in het vertaalsysteem. Het begrijpen en gebruiken van typologische verschillen tussen talen maakt het mogelijk vertaalsystemen typologisch meer robuust te maken zonder de aannames van de onderliggende vertaalmodellen drastisch te hoeven veranderen.

We beginnen met een studie van het effect van woordvolgordevrijheid op preorderen, een populaire techniek om de woordvolgorde te modelleren in phrase-based machinevertaling. We laten zien dat het gebruiken van een keuzeruimte van potentiële woordvolgorden in plaats van een enkele woordvolgorde en het inbouwen van deze ruimte in het vertaalmodel door middel van woordvolgordepermutatieroosters een principiële oplossing biedt voor het verbeteren van typologische robuustheid voor preorderen.

Vervolgens laten we zien dat phrase-based machinevertaling voor typologisch verschillende taalparen verbeterd kan worden door het verkleinen van de verschillen in morfologische complexiteit tussen bron- en doeltaal. Voor flexiemorfologie doen we

dit door het verrijken van een morfologisch arme brontaal met ongemarkeerde morfologische kenmerken, wat woordkeuze in de doeltaal verbetert. Voor samenstellingen stellen we een semantisch gemotiveerd samenstellingsmodel voor, dat samengestelde woorden in hun betekenisdragende onderdelen opsplitst. Dit stelt het vertaalsysteem in staat om met vergelijkbare vertalingseenheden in de bron- en doeltaal te opereren.

Tenslotte laten wij zien dat taaltypologie niet alleen voor het overbruggen van prestatieverschillen tussen typologisch verschillende talen van dienst is, maar dat het ook een kennisbron vormt om reorderingsmodellen te leiden en universele reorderingsmodellen voor meerdere doeltalen te vergemakkelijken. Zulke universele reorderingsmodellen kunnen op een data-gebaseerde manier leren op welke taaltypologische aspecten te letten, ze bevorderen generalisatie en ze hebben minder trainingsdata nodig dan modellen voor afzonderlijke talen.

List of Publications

Joachim Daiber, Miloš Stanojević, and Khalil Sima'an. Universal reordering via linguistic typology. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3167–3176, Osaka, Japan, December 2016b. The COLING 2016 Organizing Committee. URL http://aclweb.org/anthology/C16-1298.

Joachim Daiber, Miloš Stanojević, Wilker Aziz, and Khalil Sima'an. Examining the relationship between preordering and word order freedom in machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 118–130, Berlin, Germany, August 2016a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-2213.

Joachim Daiber and Khalil Sima'an. Machine translation with source-predicted target morphology. In *Proceedings of the 15th Machine Translation Summit (MT Summit 2015)*, pages 283–296, Miami, USA, 2015b.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. Splitting compounds by semantic analogy. In Jan Hajič and António Branco, editors, *Proceedings of the 1st Deep Machine Translation Workshop*, pages 20–28, Praha, Czechia, 2015. ÚFAL MFF UK, ÚFAL MFF UK. ISBN 978-80-904571-7-1. URL http://www.aclweb.org/anthology/W15-5703.

Joachim Daiber and Khalil Sima'an. Delimiting morphosyntactic search space with source-side reordering models. In Jan Hajič and António Branco, editors, *Proceedings of the 1st Deep Machine Translation Workshop*, pages 29–38, Praha, Czechia, 2015a. ÚFAL MFF UK, ÚFAL MFF UK. ISBN 978-80-904571-7-1. URL http://www.aclweb.org/anthology/W15-5704.

Joachim Daiber and Rob van der Goot. The denoised web treebank: Evaluating dependency parsing under noisy input conditions. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard,

Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Titles in the ILLC Dissertation Series:

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder...": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**
Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2010-12: **Martin Mose Bentzen**
Stit, Iit, and Deontic Logic for Action Types

ILLC DS-2011-01: **Wouter M. Koolen**
Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
Small steps in dynamics of information

ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access

ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities

ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure

ILLC DS-2011-07: **Jop Briët**
Grothendieck Inequalities, Nonlocal Games and Optimization

ILLC DS-2011-08: **Stefan Minica**
Dynamic Logic of Questions

ILLC DS-2011-09: **Raul Andres Leal**
Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications

ILLC DS-2011-10: **Lena Kurzen**
Complexity in Interaction

ILLC DS-2011-11: **Gideon Borensztajn**
The neural basis of structure in language

ILLC DS-2012-01: **Federico Sangati**
Decomposing and Regenerating Syntactic Trees

ILLC DS-2012-02: **Markos Mylonakis**
Learning the Latent Structure of Translation

ILLC DS-2012-03: **Edgar José Andrade Lotero**
Models of Language: Towards a practice-based account of information in natural language

ILLC DS-2012-04: **Yurii Khomskii**
Regularity Properties and Definability in the Real Number Continuum: idealized forcing, polarized partitions, Hausdorff gaps and mad families in the projective hierarchy.

ILLC DS-2012-05: **David García Soriano**
Query-Efficient Computation in Property Testing and Learning Theory

ILLC DS-2012-06: **Dimitris Gakis**
Contextual Metaphilosophy - The Case of Wittgenstein

ILLC DS-2012-07: **Pietro Galliani**
The Dynamics of Imperfect Information

ILLC DS-2012-08: **Umberto Grandi**
Binary Aggregation with Integrity Constraints

ILLC DS-2012-09: **Wesley Halcrow Holliday**
Knowing What Follows: Epistemic Closure and Epistemic Logic

ILLC DS-2012-10: **Jeremy Meyers**
Locations, Bodies, and Sets: A model theoretic investigation into nominalistic mereologies

ILLC DS-2012-11: **Floor Sietsma**
Logics of Communication and Knowledge

ILLC DS-2012-12: **Joris Dormans**
Engineering emergence: applied theory for game design

ILLC DS-2013-01: **Simon Pauw**
Size Matters: Grounding Quantifiers in Spatial Perception

ILLC DS-2013-02: **Virginie Fiutek**
Playing with Knowledge and Belief

ILLC DS-2013-03: **Giannicola Scarpa**
Quantum entanglement in non-local games, graph parameters and zero-error information theory

ILLC DS-2014-01: **Machiel Keestra**
Sculpting the Space of Actions. Explaining Human Action by Integrating Intentions and Mechanisms

ILLC DS-2014-02: **Thomas Icard**
The Algorithmic Mind: A Study of Inference in Action

ILLC DS-2014-03: **Harald A. Bastiaanse**
Very, Many, Small, Penguins

ILLC DS-2014-04: **Ben Rodenhäuser**
A Matter of Trust: Dynamic Attitudes in Epistemic Logic

ILLC DS-2015-01: **María Inés Crespo**
Affecting Meaning. Subjectivity and evaluativity in gradable adjectives.

ILLC DS-2015-02: **Mathias Winther Madsen**
The Kid, the Clerk, and the Gambler - Critical Studies in Statistics and Cognitive Science

ILLC DS-2015-03: **Shengyang Zhong**
Orthogonality and Quantum Geometry: Towards a Relational Reconstruction of Quantum Theory

ILLC DS-2015-04: **Sumit Sourabh**
Correspondence and Canonicity in Non-Classical Logic

ILLC DS-2015-05: **Facundo Carreiro**
Fragments of Fixpoint Logics: Automata and Expressiveness

ILLC DS-2016-01: **Ivano A. Ciardelli**
Questions in Logic

ILLC DS-2016-02: **Zoé Christoff**
Dynamic Logics of Networks: Information Flow and the Spread of Opinion

ILLC DS-2016-03: **Fleur Leonie Bouwer**
What do we need to hear a beat? The influence of attention, musical abilities, and accents on the perception of metrical rhythm

ILLC DS-2016-04: **Johannes Marti**
Interpreting Linguistic Behavior with Possible World Models

ILLC DS-2016-05: **Phong Lê**
Learning Vector Representations for Sentences - The Recursive Deep Learning Approach

ILLC DS-2016-06: **Gideon Maillette de Buy Wenniger**
Aligning the Foundations of Hierarchical Statistical Machine Translation

ILLC DS-2016-07: **Andreas van Cranenburgh**
Rich Statistical Parsing and Literary Language

ILLC DS-2016-08: **Florian Speelman**
Position-based Quantum Cryptography and Catalytic Computation

ILLC DS-2016-09: **Teresa Piovesan**
Quantum entanglement: insights via graph parameters and conic optimization

ILLC DS-2016-10: **Paula Henk**
Nonstandard Provability for Peano Arithmetic. A Modal Perspective

ILLC DS-2017-01: **Paolo Galeazzi**
Play Without Regret

ILLC DS-2017-02: **Riccardo Pinosio**
The Logic of Kant's Temporal Continuum

ILLC DS-2017-03: **Matthijs Westera**
Exhaustivity and intonation: a unified theory

ILLC DS-2017-04: **Giovanni Cinà**
Categories for the working modal logician

ILLC DS-2017-05: **Shane Noah Steinert-Threlkeld**
Communication and Computation: New Questions About Compositionality

ILLC DS-2017-06: **Peter Hawke**
The Problem of Epistemic Relevance

ILLC DS-2017-07: **Aybüke Özgün**
Evidence in Epistemic Logic: A Topological Perspective

ILLC DS-2017-08: **Raquel Garrido Alhama**
Computational Modelling of Artificial Language Learning: Retention, Recognition & Recurrence

ILLC DS-2017-09: **Miloš Stanojević**
Permutation Forests for Modeling Word Order in Machine Translation

ILLC DS-2018-01: **Berit Janssen**
Retained or Lost in Transmission? Analyzing and Predicting Stability in Dutch Folk Songs

ILLC DS-2018-02: **Hugo Huurdeman**
Supporting the Complex Dynamics of the Information Seeking Process

ILLC DS-2018-03: **Corina Koolen**
Reading beyond the female: The relationship between perception of author gender and literary quality

ILLC DS-2018-04: **Jelle Bruineberg**
Anticipating Affordances: Intentionality in self-organizing brain-body-environment systems