Syntactic and Semantic Features for Statistical and Neural Machine Translation

Maria Nădejde

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2018

Abstract

Machine Translation (MT) for language pairs with long distance dependencies and word reordering, such as German–English, is prone to producing output that is lexically or syntactically incoherent. Statistical MT (SMT) models used explicit or latent syntax to improve reordering, however failed at capturing other long distance dependencies. This thesis explores how explicit sentence-level syntactic information can improve translation for such complex linguistic phenomena. In particular, we work at the level of the syntactic-semantic interface with representations conveying the predicate-argument structures. These are essential to preserving semantics in translation and SMT systems have long struggled to model them.

String-to-tree SMT systems use explicit target syntax to handle long-distance reordering, but make strong independence assumptions which lead to inconsistent lexical choices. To address this, we propose a Selectional Preferences feature which models the semantic affinities between target predicates and their argument fillers using the target dependency relations available in the decoder. We found that our feature is not effective in a string-to-tree system for German→English and that often the conditioning context is wrong because of mistranslated verbs.

To improve verb translation, we proposed a Neural Verb Lexicon Model (NVLM) incorporating sentence-level syntactic context from the source which carries relevant semantic information for verb disambiguation. When used as an extra feature for reranking the output of a German→English string-to-tree system, the NVLM improved verb translation precision by up to 2.7% and recall by up to 7.4%.

While the NVLM improved some aspects of translation, other syntactic and lexical inconsistencies are not being addressed by a linear combination of independent models. In contrast to SMT, neural machine translation (NMT) avoids strong independence assumptions thus generating more fluent translations and capturing some long-distance dependencies. Still, incorporating additional linguistic information can improve translation quality.

We proposed a method for tightly coupling target words and syntax in the NMT decoder. To represent syntax explicitly, we used CCG supertags, which encode subcategorization information, capturing long distance dependencies and attachments. Our method improved translation quality on several difficult linguistic constructs, including prepositional phrases which are the most frequent type of predicate arguments. These improvements over a strong baseline NMT system were consistent across two language pairs: 0.9 BLEU for German→English and 1.2 BLEU for Romanian→English.

Lay Summary

Machine Translation (MT) is the task of translating a sentence written in some "source" language into another "target" language, automatically, using computer algorithms. MT systems have progressed significantly in the last fifty years from early rule-based systems, involving hand crafted translation rules designed for each language by experts, to systems based on neural networks, capable of learning complex linguistic phenomena from a large corpus of translated sentences. Despite the progress, MT for language pairs with long distance dependencies and word reordering, such as German–English, is prone to producing output that is lexically or syntactically incoherent.

This thesis claims that explicit sentence-level syntactic context is required from both the source-side and the target-side to improve machine translation when long distance dependencies are involved. In particular, we work at the level of the syntactic-semantic interface with representations conveying the predicate-argument structures. These are essential to preserving semantics in translation and MT systems have long struggled to model them.

We first augment a syntax-based statistical MT system with a Selectional Preferences feature modeling the semantic affinities between target predicates and their argument fillers. We found that our feature is not improving the system for German→ English and that often the conditioning context is wrong because of mistranslated verbs. To improve verb translation, we proposed a Verb Lexicon model incorporating syntactic context from the source sentence which carries relevant semantic information for verb disambiguation. While this model improved some aspects of translation, other syntactic and lexical inconsistencies still occurred. This happens because the syntax-based MT system does not have a global representation of the source and target sentence.

In contrast to syntax-based MT, MT systems based on neural networks (NMT) are able to learn representations of the entire source sentence and translation history, which capture some long distance dependencies. Still, incorporating additional linguistic information can improve translation quality. We proposed a method for tightly coupling target words and syntax in NMT, and showed it improves machine translation quality, on several difficult linguistic constructs, for German→English and Romanian→English. We used a syntactic representation encoding the subcategorization frame of predicates, which helped improve translation of prepositional phrases, the most frequent type of predicate arguments.

Acknowledgements

I would like to thank my supervisors, Philipp Koehn and Alexandra Birch, for their support and guidance during my PhD. Thanks to Philipp, I had the opportunity to work on my PhD at the University of Edinburgh, with some of the best researchers in MT, to visit the CLSP group at Johns Hopkins University and to travel across two continents while attending conferences. Back in Bucharest I never dreamt that one day I would be doing all these things. I owe a big thank you to Lexi, who has continuously encouraged me to push forward, has helped transform rough drafts into conference papers and has been an invaluable source of positive energy. I would not have managed to finish this thesis without her support. I am also thankful to the thesis committee, Bonnie Webber and Ondřej Bojar, for their suggestions on improving this thesis.

I received a lot of help from other researchers throughout my PhD, some of them anonymous through double-blind peer-reviewing. I want to thank Rico Sennrich for all the discussions and advice related to my work, and also for his help with the Nematus toolkit and RDLM. I am indebted to Philipp Williams and Hieu Hoang for their amazing work on implementing syntax-based models in the Moses toolkit. I also appreciate the help and clarifications regarding CCG from Siva Reddy and Bharat Ram Ambati.

My time in Edinburgh as a PhD student was much happier thanks to the wonderful people in the MT group: Nikolay Bogoychev, Christian Buck, Marcin Junczys-Dowmunt, Tomasz Dwojak, Federico Fancellu, Ulrich Germann, Liane Guillou, Barry Haddow, Eva Hasler, Kenneth Heafield, Hieu Hoang, Matthias Huck, Adam Lopez, David Matthews, Antonio Miceli, Miles Osborne, Hérve Saint-Amand, Rico Sennrich, Chara Tsoukala, Dominikus Wetzel and Philip Williams. We had interesting conversations, shared laughs and drank pints and pints of room temperature beers. I will miss the company of Annie, Cristina, Lea, Spandana, Stella, Andrew, Bharat, Des, Philipp, Siva and many others from the Informatics Forum. I will also hold fond memories of Casa Maria, which I shared with Maria Alecu.

I am especially grateful to Aurora Constantin and Mihaela Dragomir for taking me into their home and being my extended family. They have made many grey Scottish days feel warmer and sunnier to me. I owe special thanks to Tassilo Barth, who patiently proofread my work, annotated errors in English-German translations and who made the last couple of years more fun and exciting.

Most of all, I appreciate the help, support, patience and love that my parents, Nadia and Andrei, have always offered me.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Maria Nădejde*)

Table of Contents

Chapter 1

Introduction

To correctly render the meaning of a source sentence, a translation requires syntactic and lexical cohesion. This is harder to achieve with machine translation for language pairs which exhibit long distance dependencies and word reordering, such as German–English. This thesis claims that the syntactic structure of the sentence is required on both the source-side and the target-side to improve machine translation in particular when long distance dependencies[1] are involved.

We test empirically three hypotheses. Our first hypothesis, that target-side semantic affinities can improve translation of predicate-argument structures, is not confirmed by empirical results. Our second hypothesis, that the source-side syntactic context improves translation of verbs, is confirmed for the German→English language pair by an increase in verb recall and precision. However, this is attained at the cost of a small decrease in overall translation quality as measured by BLEU , an automatic metric. Our third hypothesis is that explicitly modeling target language syntax in neural machine translation improves translation quality. This is confirmed empirically, for German→English, a high-resource pair, and for Romanian→English, a low-resource pair. The improvements are consistent across several difficult linguistic constructs, including prepositional phrases which are the most frequent type of predicate arguments.

1.1 Syntax for Machine Translation

Statistical machine translation (SMT) has been successfully applied to many language pairs and domains. An SMT system breaks down the translation of a sentence into the

[1] We consider a dependency relation between a head and dependent which are separated by several words and which may not fall within the n-gram language model context, to be long-distance.

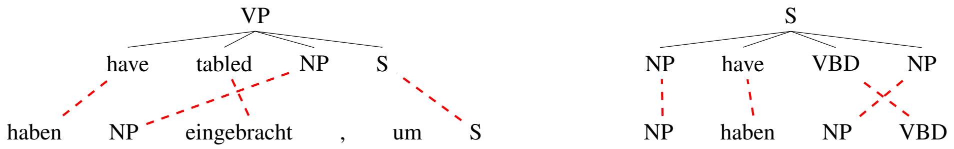
translation of basic independent sub-units, with some attached translation probability. Phrase-based SMT systems use phrase-pairs as basic translation units, which capture local movement of words within a phrase-pair, multi-word expressions and idioms. However, when unconstrained movement of phrases is allowed, searching for the optimal ordering becomes an NP-complete problem under a phrase-based model [Knight, 1999].

Word order differences between languages account for most of the variation in translation performance for phrase-based SMT models [Birch, 2011]. Studying the characteristics of European language pairs [Birch, 2011] also showed that the German→English translation direction involves the most reordering among eleven language pairs with the same target language. For this reason, we test all of our hypotheses on this language pair. German allows verbs to appear in different positions: in perfect tense the main verb appears at the end of the sentence and some verbs have separable particles that are placed at the end of the sentence. Phrase-based SMT struggles to handle word reordering for this language pair and does not handle long-distance reordering, exemplified next. In Figure 1.1 we give an example of a translation for the German→English language pair that requires two verbs to be re-ordered. The first reordering involves the verb *"eingebracht"*, and its direct object *"einen Gesetzesvorschlag"*. The translation of the second verb, *"etablieren"*, requires a long distance movement as it needs to be placed at the beginning of the second clause. While the first reordering could be performed by phrase-based models, they would not handle the second long distance movement and would potentially drop one of the verbs. In contrast, string-to-tree SMT system can handle both cases as shown in the example translation provided in Figure 1.1.

Source	Die Kongress Abgeordneten haben einen Gesetzesvorschlag **eingebracht**, um die Organisation von Gewerkschaften als Bürgerrecht **zu etablieren**.
Gloss	Congressmen have a legislation **proposed**, of the organization of trade unions as civil right **to establish**.
Reference	Congressmen have **proposed** legislation **to protect** union organizing as a civil right.
String-to-Tree SMT	Congressmen have **tabled** a bill **to establish** the organization of trade unions as a civil right.

Figure 1.1: Example of reordering by a German-to-English statistical machine translation (SMT) model.

String-to-tree SMT systems use synchronous context free grammar (SCFG) [Aho and Ullman, 1969] rules, with syntactic annotation on the target side, to handle long distance re-ordering and produce syntactically well-formed translations. Modeling the target-side syntax is important for machine translation since a syntactically well-formed sentence is more fluent and potentially more accurate. In Figure 1.2 we give examples of SCFG rules which can reorder the verb *"eingebracht"* and its argument, according to the target word order. The figure shows a target sub-tree with the alignment between the target non-terminals and the corresponding source spans. String-to-tree translation rules have generic (X) non-terminal labels on the source-side that correspond one-to-one with syntactic non-terminal labels on the target side. The target-side non-terminals are either part-of-speech labels or phrase structure labels and the mapping between source and target spans is indicated in the SCFG rule by the subscript numbers.



VP → have tabled NP$_0$ S$_1$ ||| haben X$_0$ eingebracht um X$_1$

S → NP$_0$ have VBD$_1$ NP$_2$ ||| X$_0$ haben X$_2$ X$_1$

Figure 1.2: Alternative synchronous context free grammar (SCFG) rules for reordering the verb *"eingebracht"* and its NP argument. The target syntactic sub-tree and the alignment of the non-terminals to the source-side spans are depicted at the top. The corresponding SCFG rule is depicted at the bottom.

These structured translation rules allow reordering by abstracting away from the lexical realization of the different syntactic constituents. The noun-phrase (NP) arguments will be translated independently of the verb by subsequent SCFG rules rooted in an NP non-terminal. By making these strong independence assumptions and limiting the lexical context, a string-to-tree system will induce translation errors such as incoherent lexical choices and missing words. In the previous example, the verb is translated as *"tabled"* which has opposite meaning in American and British English, while the intended and non-ambiguos translation is *"proposed"* or *"introduced"*. Even a strong string-to-tree system for German→English only retains about 66 percent of the meaning of the source semantic frames [Birch et al., 2013].

Previous work augmented string-to-tree systems with either global source or tar-

Verb	Relation	Argument	SelAssoc
see	dobj	PRN	0.123
		movie	0.022
		episode	0.001
is–hereditary	nsubj	disease	0.267
		monarchy	0.148
		title	0.082

Table 1.1: Examples of *selectional association* (SelAssoc) scores for different verbs. PRN is the class of pronouns. PRN is the part-of-speech tag for pronouns, dobj is direct object, nsubj stands for subject.

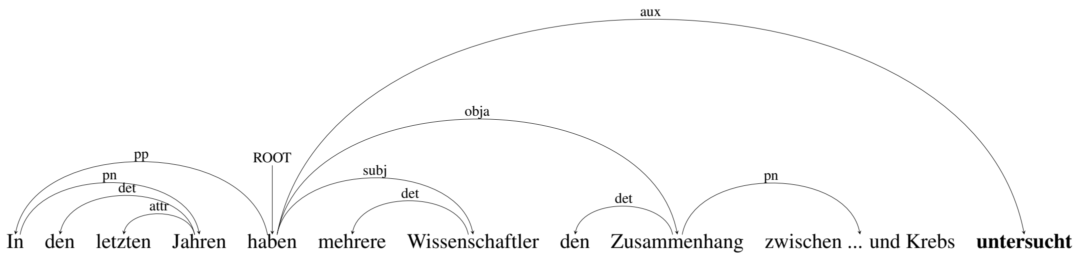
get information. For example, Weller et al. [2014] annotate SCFG rules translating prepositional-phrases with noun-class information and [Sennrich, 2015] propose a language model over syntactic n-grams. However, these models do not focus on the lexical semantic affinities between target predicates and their arguments. To improve rule selection for systems based on formal SCFGs[2], some proposed discriminative models integrating a wider source context than is available in typical translation units [Braune et al., 2015, 2016, Liu et al., 2008]. Still, these models rely more on structural differences between the target-side of SCFG rules, neglecting the lexical selection of verbs.

In this thesis, we propose softening the independence assumptions of string-to-tree systems and improving lexical cohesion by incorporating global source and target syntactic context. We explore two initial hypotheses: that knowledge of target-side semantic affinities improves lexical cohesion of predicate-argument pairs and that global source-side syntactic context improves lexical choices for verbs.

To achieve lexical cohesion at the level of the predicate-argument structure, we introduce a feature to model target-side selectional preferences of predicates. Selectional preferences describe the semantic affinities between predicates and their argument fillers. For example, the verb *"drinks"* has a strong preference for arguments in the conceptual class of *"liquids"*. Therefore, the word *"wine"* can be disambiguated when it appears in relation to the verb *"drinks"*.

A corpus driven approach to modeling selectional preferences usually involves extracting triples of (*syntactic relation, predicate, argument*) and computing co-occurrence statistics. Our feature is based on the *selectional association* measure proposed by

[2] SCFG rules without explicit syntactic annotation and with generic non-terminal labels X.


aux
obja
pp
ROOT
subj
pn
det
attr
det
det
pn
In den letzten Jahren haben mehrere Wissenschaftler den Zusammenhang zwischen ... und Krebs untersucht

Reference:	In recent years , a number of scientists have **studied** the links [between ... and cancer]						
Window context	und	Krebs	**untersucht**	</s>	</s>		
Syntactic context	source verb	parent	dependents			pp modifier	
	untersucht	haben	Wissenschaftler	Zusammenhang	<null>	in	Jahren

Figure 1.3: Example of window and syntactic context extracted for the source verb untersucht (studied; highlighted in bold).

Resnik [1996] which follows this approach. We give examples of the *selectional association* scores for different verbs and their arguments in Table 1.1. The verb *"see"* takes on many arguments as direct objects and therefore has lower selectional association scores for the arguments in this syntactic relation. In contrast, the predicate *"is-hereditary"*[3] takes on fewer arguments for which it has stronger selectional association scores.

Our first hypothesis, that target-side semantic affinities can improve translation of predicate-argument structures, is not confirmed by empirical results. According to the analysis presented in Chapter 4, as the distance between predicates and arguments increases and the target syntactic structure becomes more complex, the translation precision decreases drastically. Furthermore, verbs are often mistranslated which negatively impacts the proposed selectional preferences feature. The analysis of verb translation in string-to-tree systems, presented in Chapter 5, shows that 20 percent of the main verbs are translated without lexical context and verb translation recall is as low as 45.5 percent. Therefore, we address the problem of verb translation in string-to-tree systems by incorporating the source verb context extracted from the syntactic structure of the source sentence. Since the syntactic context is extracted from the source sentence we can include most of the verb's dependents, in particular the core arguments that carry most semantic information relevant to verb disambiguation.

[3] Here the verb *"is"* has the role of an auxiliary while the adjective *"hereditary"* is the semantic predicate. In this case, the Stanford parser will attach the arguments to the semantic predicate.

Source	Oder wollen Sie herausfinden , **über** was andere reden ?
Reference	Or do you want to find out what others are talking **about** ?
NMT	Or would you like to find out **about** what others are talking **about** ?

Figure 1.4: Example of neural machine translation (NMT) of a German→English sentence involving a subordinate clause, compared to gold standard reference. Incorrectly translated phrase highlighted in bold.

We propose a verb specific lexicon model with the knowledge that verbs have the most outgoing dependency relations, are central to semantic structures and therefore would benefit most from a source-side syntactic context. In Figure 1.3 we give an example of how we can extract all the relevant lexical context of the source verb by following its syntactic dependency relations. In contrast, a window context centered on the source verb provides only one content word which is not an argument of the verb. Our second hypothesis, that the source-side syntactic context improves translation of verbs, is confirmed for the German→English language pair by an increase in verb recall and precision. However, this is attained at the cost of a small decrease in overall translation quality as measured by BLEU , an automatic metric.

Although string-to-tree systems with explicit target syntax out-perform phrase-based systems for syntactically divergent language pairs such as German–English, both systems suffer from data sparsity and strong independence assumptions. Neural machine translation (NMT) models address both these issues by learning distributed representations of the source and target words and by modeling the entire source context and target history when generating a translation. These are desirable properties when trying to model long-distance dependencies and re-ordering. It has been shown that NMT models are able to partially learn source-side syntactic information from sequential lexical information. However, some complex syntactic phenomena such as prepositional phrase attachment are poorly modeled [Shi et al., 2016, Bentivogli et al., 2016]. In Figure 1.4 we give an example of a translation, produced by an NMT system, which is locally fluent but does not capture the correct syntactic structure. The system generates two fluent constructs in the target language involving the preposition *"about"*: *"find out about"* and *"talking about"*. However, the syntactic structure of the question requires only the second occurrence of the preposition.

Previous work has attempted to induce structure when modeling the source sentence by using convolutional neural networks [Kalchbrenner and Blunsom, 2013, Cho et al., 2014a] or syntactically guided attentional recurrent networks [Eriguchi et al., 2016]. Others have improved translation quality by incorporating explicit source-side linguistic features [Luong et al., 2016, Sennrich and Haddow, 2016]. Applying target-side linguistic factors in NMT, namely morphological tags, has also been briefly investigated [Martínez et al., 2016]. However, no previous work has explored the more general problem of including target syntax in NMT: comparing tightly and loosely coupled syntactic information and showing source and target syntax are complementary.

We propose a third hypothesis, that explicitly modeling target language syntax in neural machine translation improves translation quality. In particular we investigate the following research questions: 1) Is tight integration of words and syntax better than multitask training? 2) Does target syntax provide complementary information to source syntax for NMT?

We present empirical results showing that explicitly modeling target-syntax improves machine translation quality, in particular on several difficult linguistic constructs, for German→English, a high-resource pair, and for Romanian→English, a low-resource pair. Furthermore, a tight coupling of words and syntax improves translation quality more than multitask training. While both approaches allow the target syntactic information to impact all parameters of the model during training, only the former approach makes the probability of the target words conditioned on target-syntax. We obtain further improvements in translation quality by combining target-syntax with source-syntax, showing that the two are complementary.

1.2 Contributions

The contributions of this thesis are:

- We explore different methods for improving robustness of string-to-tree systems and build a state-of-the art system for German→English.
- We propose a Selectional Preferences Model which captures semantic affinities between target predicates and their arguments. We show that the model is not effective when used as a feature in a string-to-tree systems for German→English, because of overlap with the language model and because of mistranslated verbs.
- We present an analysis of verb translation in string-to-tree systems for German→English highlighting that verb translation recall is as low as 45% and that 20% of the main verbs are translated without lexical context.

- We propose a Neural Verb Lexicon Model to address the problem of mistranslated verbs in string-to-tree systems. The model uses a rich source-side syntactic context, including the subcategorization frame, improving verb translation precision by up to 2.7% and recall by up to 7.4%.
- We propose a novel method to incorporate explicit target-syntax in a neural machine translation system, by interleaving target words with their corresponding combinatory categorial grammar (CCG) supertags. We show that target language syntax improves translation quality in both high-resource and low-resource scenarios, and that a tight coupling of target words and syntax (by interleaving) is better than a loose coupling as in multitask learning.
- We show that combining our method for Syntax-aware NMT (SNMT) with target CCG supertags with a complementary framework incorporating source-side linguistic information, yields additional improvement in translation quality.
- We present a fine-grained analysis of SNMT and show consistent gains for different linguistic phenomena and sentence lengths.

1.3 Thesis Outline

In this section we outline the structure of this thesis and specify which parts relate to the contributions.

Chapter 2 presents background on statistical machine translation (SMT) and neural machine translation (NMT). In Section 2.1 we present a brief overview of SMT. In Section 2.2 we introduce string-to-tree SMT systems, followed by an overview of neural language models in Section 2.3 and finally we describe the state-of-the-art neural machine translation models in Section 2.4.

In Chapter 3 we explore several methods of improving the robustness of string-to-tree systems for translating into English. In section 3.2 we describe the details of training a competitive baseline string-to-tree system. Then in section 3.3 we present three ways of improving grammar coverage: tree restructuring, realigning verbs and pre-processing named entities. We conclude this chapter with Section 3.4, an error analysis of a string-to-tree system for German→English, which highlights aspects that can be improved using global source and target syntactic context.

In Chapter 4 we explore whether a Selectional Preferences feature, which captures semantic affinities between the target predicates and their argument fillers, is useful for translating ambiguous predicates and arguments. In Section 4.2 we introduce the problem of translating ambiguous predicate-argument structures with string-to-tree systems and prior work that addresses this. In Section 4.3 we present a contrastive evaluation of syntactic representations showing that a string-to-tree system with target-side dependency relations is competitive with the string-to-tree system with target-side phrase-structures introduced in the previous chapter. In Section 4.4 we formally describe the selectional preference feature for dependency-based string-to-tree systems. Section 4.5 describes the experimental setup and Section 4.6 presents the results of the automatic evaluation, as well as a qualitative analysis of the machine translated output.

In Chapter 5 we address the problem of verb translation in string-to-tree systems and propose a Neural Verb Lexicon Model which uses the source-side syntactic context to improve the lexical choice for verbs. In Section 5.2 we exemplify why verb translation is problematic for string-to-tree systems and contrast our proposed model with prior work on discriminative word lexicon and rule selection models. In Section 5.3 we present a thorough analysis of verb translation conducted on a German→English string-to-tree system, and we determine to what extent this is a problem for the state-of-the-art system. Section 5.4 describes our proposed neural verb lexicon model and presents ablation experiments evaluated in terms of verb prediction accuracy. Finally, in Section 5.5, we investigate whether the verb lexicon model is able to improve translation quality by integrating the model as an additional feature for re-reranking the output of the string-to-tree system.

In Chapter 6 we examine the benefit of incorporating sentence-level syntactic information on the target-side of NMT. We propose a method for tightly coupling words and syntax by interleaving the target syntactic representation, in the form of CCG supertags, with the word sequence. In Section 6.2 we discuss the limitations of NMT and previous work on integrating source or target syntactic information. In Section 6.3 we describe the syntactic representation and different strategies of coupling it with the translated words in the decoder or in the encoder of the NMT system. In Section 6.4 we describe the experimental setup and training parameters for the NMT systems. In Section 6.5 we evaluate the effect of target syntax on overall translation quality and make a finer grained analysis with respect to different linguistic constructions and sentence lengths.

In Chapter 7 we summarize our contributions to the field of machine translation and present future research directions.

1.4 Related Publications

In Chapter 3 we present methods for improving robustness of string-to-tree systems, previously reported in system description papers submitted at several evaluation campaigns organized by the Workshop on Statistical Machine Translation (WMT) [Nădejde et al., 2013, Williams et al., 2014, 2015, 2016]. My contribution to the cited papers was to design, conduct, evaluate and report the experiments for the German→English and Romanian→English language pairs. In this chapter, we also refer to the HMEANT evaluation results from Birch et al. [2013], comparing the string-to-tree system with phrase-based and rule-based systems. My contribution to this paper was threefold: training the string-to-tree system for German→English that was evaluated, manually evaluating translations for German→English with the HMEANT annotation tool and compiling the documentation on using the HMEANT annotation tool.

The Selectional Preferences model and experimental results presented in Chapter 4 were previously published under the title "*Modeling selectional preferences of verbs and nouns in string-to-tree machine translation*" in the Proceedings of the First Conference on Machine Translation (WMT 2016) [Nădejde et al., 2016a]. I designed, conducted, evaluated and reported all the experiments in the cited paper. The comparison of syntactic representations in string-to-tree systems was reported in the WMT 2016 system description paper [Williams et al., 2016]. My contribution to the cited paper was to design, conduct, evaluate and report the experiments for the German→English and Romanian→English language pairs, including those comparing syntactic representations in string-to-tree systems.

The Neural Verb Lexicon Model and experimental results presented in Chapter 5 were previously published under the title "*A Neural Verb Lexicon Model with Source-side Syntactic Context for String-to-Tree Machine Translation*" in the Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2016) [Nădejde et al., 2016b]. I designed, conducted, evaluated and reported all the experiments in the cited paper.

In Chapter 6 we presented an extended version of the work published under the title "*Predicting Target Language CCG Supertags Improves Neural Machine Translation*" in the Proceedings of the Second Conference on Machine Translation (WMT 2017) [Nădejde et al., 2017]. An earlier version of this work was uploaded to the pre-print server arXiv under the title "*Syntax-aware Neural Machine Translation Using CCG*". I designed, conducted, evaluated and reported all the experiments in the cited paper. Siva Reddy provided advice and examples regarding CCG supertags, as well as the rules for grouping sentences according to linguistic constructs. Rico Sennrich provided the code for the multitask framework in the Nematus toolkit which I used for a subset of the experiments. Tomasz Dwojak partially implemented the “distinct softmax” framework, which I finalized and used for one of the experiments. All the co-authors contributed with valuable suggestions and feedback.

Chapter 2

Background

2.1 Introduction

Machine Translation (MT) is the task of translating a sentence written in some "source" language into another "target" language, automatically, using computer algorithms. MT systems have progressed significantly in the last fifty years from early rule-based systems, involving hand crafted translation rules designed for each language by experts, to systems based on neural networks, capable of learning complex linguistic phenomena from a large corpus of translated sentences and without additional linguistic annotation.

Statistical Machine Translation (SMT) systems are data-driven, using statistics about basic translation units collected from a large sentence-aligned corpus of translations, and are applicable to all language pairs for which such a corpus is available. Phrase-based SMT systems use phrase-pairs as basic translation units, which are learned in an unsupervised manner by first inducing a word alignment and then applying heuristics to group aligned words into aligned phrases. In Figure 2.1 we show a possible phrase-pair alignment that can be extracted from this example.

The phrase-based translation model attaches a probability to all extracted phrase-pairs, $p_{TM}(\bar{f}_i|\bar{e}_i)$. The translation of a sentence can be found by searching for the most probable combination of phrase-pairs covering the entire source sentence. The probability of the translation e conditioned on the source sentence f can be computed using a discriminative log-linear model [Och and Ney, 2002] based on K feature functions $h_k(e, f)$ including the translation model (TM) and a language model (LM):

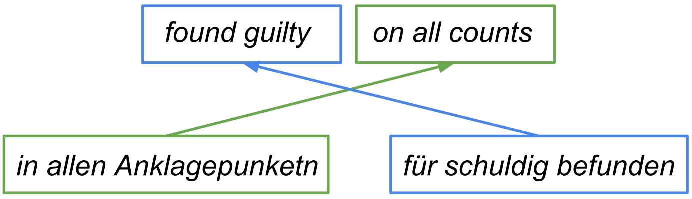


Figure 2.1: Example of phrase-pairs extracted from a German-English sentence pair by consolidating word alignments. Phrase-pair boundaries are indicated by the boxes and source-target correspondence is marked by arrows.

$$\begin{aligned} P(e|f) &= \exp(\sum_{k=1}^{K} \lambda_k\, h_k(e,f)) \quad (2.1)\\ &= \exp(\lambda_{TM} \sum_{i=1}^{I} \log p_{TM}(\bar{f}_i|\bar{e}_i) \quad (2.2)\\ &+ \lambda_{LM} \sum_{i=1}^{T} \log p_{LM}(e_i|e_1,...,e_{i-1}) + ... + \lambda_K h_K(e,f)) \quad (2.3) \end{aligned}$$

Phrase-based SMT systems are good at capturing local reorderings, multi-word expressions, idioms and other non-syntactic phrases. However, they do not model long-distance dependencies and reordering which are often needed in some language pairs to correctly render morphological and syntactic phenomena in the target language. Furthermore, when unconstrained movement of phrases is allowed, searching for the optimal ordering becomes an NP-complete problem [Knight, 1999].

Several SMT models that are formally syntactic or purely syntax driven have been proposed in the past years in an effort to include more structural and linguistic information into SMT. Hierarchical SMT (Hiero) models [Chiang, 2005] allow phrase-pairs with gaps which can generate long distance re-ordering and are formally weighted synchronous context free grammars (SCFG) [Aho and Ullman, 1969]. The basic translation units are rewrite rules with generic non-terminals X where the right-hand side consists of source (α) and target (γ) pairs that are aligned.

$$X \rightarrow \langle \gamma,\ \alpha,\ \sim \rangle \qquad (2.4)$$

where X is the only non-terminal label, γ, α are string of terminals and non-terminals and $\sim$ is the alignment between source and target non-terminals. Based on the word alignment in Figure 2.1, the following hierarchical phrase pairs could be extracted that allow the reordering of the prepositional phrase and the verb phrase. The alignment between non-terminals is indicated by the subscript indices. The source-side non-terminals correspond one-to-one with the target-side non-terminals:

$$X \rightarrow \langle on\ all\ counts,\ in\ allen\ Anklagepunkten \rangle \tag{2.5}$$

$$X \rightarrow \langle found\ guilty\ X_1,\ X_1\ f\ddot{u}r\ schuldig\ befunden \rangle \tag{2.6}$$

The Hiero decoding algorithm is an extension to chart-parsing and recursively expands all non-terminals in a rule with other rules that match the source context. The non-terminal X_1 in the second rule can be expanded using the first rule, to translate the entire verb phrase.

Hiero models have broad coverage as there are no constraints on the discontinuous phrases. However, this leads to over-generalization. String-to-tree systems address this problem with a fine-grained set of target-side non-terminals which correspond to syntactic constituents and are linguistically annotated. This leads to grammatically well-formed translations which are more fluent, and syntactically motivated word reordering. Instead of the generic non-terminal label X, the target-side non-terminal will have labels corresponding to phrase structures such as VP for verb phrase and PP for prepositional phrase:

$$VP \rightarrow \langle found\ guilty\ PP_1,\ X_1\ f\ddot{u}r\ schuldig\ befunden \rangle \tag{2.7}$$

String-to-tree models have been successfully applied to language pairs exhibiting long-distance word reordering such as Chinese-English and German-English. However, these models make strong independence assumptions resulting in errors such as syntactically and semantically incoherent verb frames. In contrast, neural machine translation (NMT) does not make strong independence assumptions thus generating more fluent translations and capturing some long-distance dependencies.

In the rest of the chapter we describe string-to-tree models in more detail (Section 2.2) as well as the main ideas behind neural networks for machine translation: neural language models (Section 2.3) and neural translation models (Section 2.4).

2.2 String-to-Tree Statistical Machine Translation

This section aims to offer an overview of the string-to-tree SMT model on which we base the work presented in Chapters 3, 4, and 5. This model is implemented in the Moses toolkit [Koehn et al., 2007] and has been used for building syntax-based systems at the University of Edinburgh.

The string-to-tree model discussed in this section was initially proposed by Galley et al. [2004a] and then further refined by Galley et al. [2006a] as well as Williams and Koehn [2012]. Unlike Hiero models, string-to-tree models require target sentences to be annotated with phrase structure trees in order to extract a SCFG with syntactic non-terminal labels. The syntactic constraints help string-to-tree systems produce more grammatical output, but on the other hand rule out non-syntactic phrase-pairs resulting in lower model coverage.

Fox [2002] carried out an analysis of phrasal cohesion and found several cases for the French-English language pair where alignments cross constituents, creating non-syntactic phrase-pairs. Such cases arise, for example, because of embedded verb phrases and the movement of adverbs. Figure 2.2 gives an example where the verb phrase “für schuldig befunden” and its translation “found guilty”, will not be extracted as a valid syntactic phrase-pair because the target side is not covered by a constituent. The same happens in the case of the modal construction “could face” and its translation “droht”. A solution to this problem, shown to significantly increase model coverage, is to restructure the parse trees prior to rule extraction. We discuss tree restructuring later in this section.

2.2.1 GHKM Rule Extraction Algorithm

The GHKM algorithm (**G**alley **H**opkins **K**night **M**arcu, Galley et al. [2004a, 2006a]) extracts SCFG rules from a word-aligned parallel corpora for which the target sentences are syntactically annotated. We describe this algorithm next, as it is the foundation of the string-to-tree models which we use in this thesis.

2.2.1.1 Minimal rules

Given a source string S, a target parse tree T and the alignment A between the source words and the leaves of the target tree, an *alignment graph* G can be defined. Figure 2.3 gives an example of an *alignment graph*. Formally the structure is a rooted,

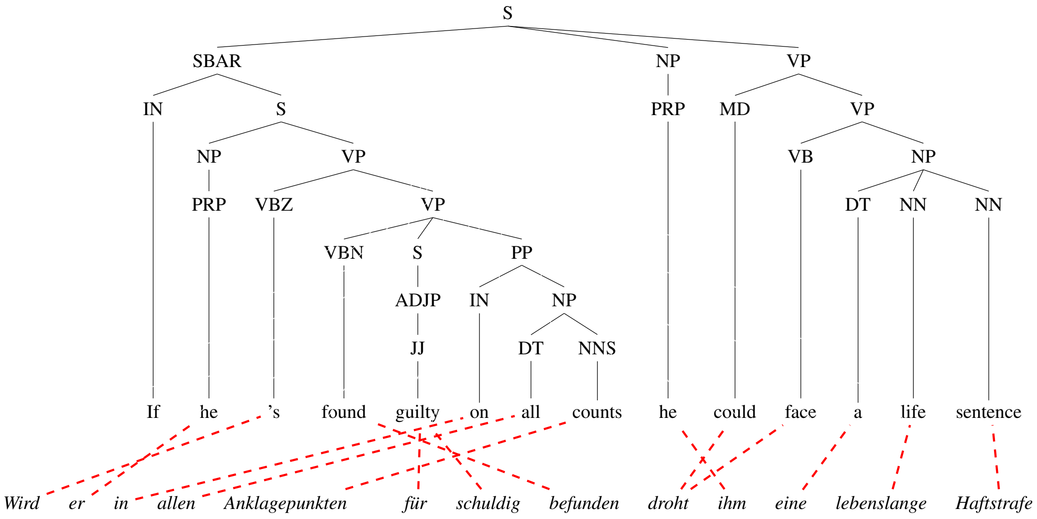


Figure 2.2: Example of non-constituent phrase-pairs that will not be extracted as SCFG rules: *"für schuldig befunden"* → *"found guilty"* and *"droht"* → *"could face"*.

directed, acyclic graph constructed from T by attaching a node for each word in S, and adding an edge between a leaf of T and a node in S if A contains an alignment for the corresponding target and source words.

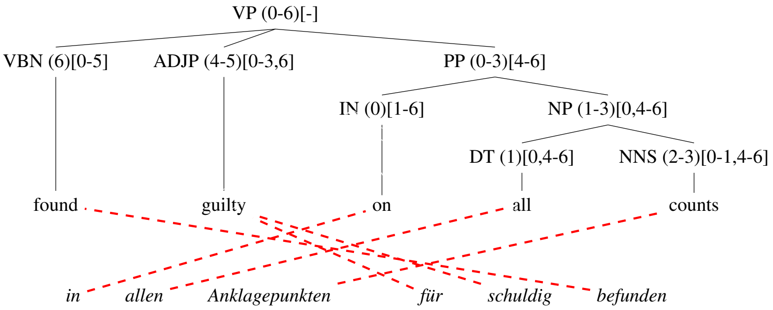


Figure 2.3: Example of an alignment graph between the source sentence (bottom) and the target sentence's phrase-structure tree (top). The non-terminal labels are marked with the node's span, in round brackets, and its complement span, in square brackets.

Each node n in T is labelled with its *span* and *complement span*. The *span* is defined by the indices of the first and last words in S that are covered by n. The *complement span* is the union of the *spans* of all other nodes n' that are neither descendants nor ancestors of n [Galley et al., 2006a]. In Figure 2.3 the spans are marked with round brackets and the complement spans with square brackets. For example the span of node PP is (0-3) since it covers the source words "in allen Anklage Punkten" and the

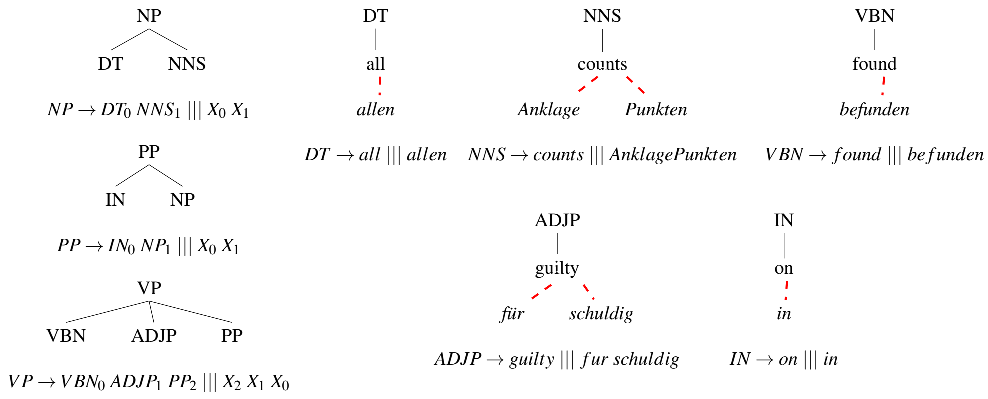


Figure 2.4: The minimal graph fragments extracted from the alignment graph in Figure 2.3. The minimal rules extracted are written under the corresponding minimal graph fragments.

complement span is [4-6] computed as the union of the nodes VBN(6) and ADJP(4-5). If for some reason the alignment would indicate that the PP node also covers the source word “für”, then the span of PP would become (0-4) and it would overlap the complement span [4-6].

A *frontier set* is defined as the set of nodes *n* in *G* for which their *span* and *complement span* do not overlap. In Figure 2.3 all nodes are in the frontier set. The intuition behind constructing the frontier set is that frontier nodes can be ordered by their spans, since the spans are contiguous and non-overlapping. For example the PP node comes before the ADJP node in the span induced order. A *frontier graph fragment* has its root and all its children nodes in the *frontier set*. For each node *n* in the frontier set there is a unique minimal frontier graph fragment rooted in *n*, which is a subgraph of all other frontier graph fragments rooted in *n*. The minimal frontier graphs extracted from the alignment graph from Figure 2.3 are shown in Figure 2.4. Finding the frontier set and minimal frontier graphs can be done in linear time. The minimal frontier graph fragments can be composed to reconstruct all other frontier graph fragments.

For each frontier node, a *minimal rule* can be extracted from the corresponding minimal frontier graph fragment using the ordering induced by the spans. The root node of the minimal frontier graph becomes the left hand side of the rule, and each frontier node is assigned a variable. The right-hand side of the rule will have a target side and a source side. The target side of the rule is built by writing the child nodes in pre-order[1]. The source side is built by substituting the source words with the variables

[1] The child nodes are traversed from left to right.

corresponding to the frontier nodes spanning them. Therefore, the target side will have the variables written in pre-order and the source side in the order induced by the spans of the frontier nodes. This way, rules model reordering but they will not try to order constituents that cannot be ordered because of the alignment. *Minimal rules* cover the training example and are consistent with the alignment. In Figure 2.4 the minimal rules are written under the minimal frontier graph they correspond to. The alignment between the source and target variables is indicated by the indexes, which correspond one-to-one, and the corresponding source side labels are a generic X. Note that the order of the source and target indexes is different in the rule rooted in the VP node. This marks that the three spans, corresponding to the verb, adverb and prepositional phrase, are reordered in the target. This reordering can also be identified in Figure 2.3 by the crossing alignment lines.

Unaligned words can be integrated in the alignment graph either heuristically, for example by always choosing highest attachment, or considering attachment points to all constituents. The later solution, proposed by Galley et al. [2006a], encodes all the possible attachments of unaligned words in a *derivation forest*.

In addition to the GHKM rules extracted from the parallel data, the grammar of a string-to-tree system also includes *glue rules*. The glue rules allow the system to concatenate partial trees during decoding, although the concatenation might not form a valid constituent. The necessity to use glue rules arises either because of computational reasons, or because words that are unknown at test time still have to be part of the derivation. Glue rules allow the following special non-terminals: $< s >$ and $< /s >$ for beginning and end of sentence, and Q as initial non-terminal. In Table 2.1 we list the types of glue rules allowed by the grammar.

Initial rule:	$Q \rightarrow < s > \ Q_0 \ ||| \ < s > \ X_0$
Final rule:	$Q \rightarrow Q_0 \ < /s > \ ||| \ X_0 \ < /s >$
Top rule:	$Q \rightarrow < s > \ NT_0 \ < /s > \ ||| \ < s > \ X_0 \ < /s >$
Generic glue rule:	$Q \rightarrow Q_0 \ NT_0 \ < /s > \ ||| \ X_0 \ X_1$

Table 2.1: The types of glue rules included in the grammar for concatenating partial derivations. NT can be any syntactic non-terminal label.

2.2.1.2 Composed rules

Two or more minimal rules that are in a parent-child relationship can be composed together to obtain larger rules with more syntactic context. Galley et al. [2006a] show that larger composed rules capture interesting linguistic phenomena, improve model coverage and significantly improve the BLEU score as compared to using only minimal rules.

For example, the following minimal rules can be compose to obtain the rule in (2.7)[2]:

$$VBN \rightarrow found \; ||| \; befunden \tag{2.8}$$

$$ADJP \rightarrow guilty \; ||| \; für \; schuldig \tag{2.9}$$

$$VP \rightarrow VBN_0 \; ADJP_1 \; PP_2 \; ||| \; X_2 \; X_1 \; X_0 \tag{2.10}$$

Analyzing the rule table, Galley et al. [2006a] observe that in the case of Chinese, composed rules are able to correctly select subject before verb ordering, while minimal rules only capture subject after verb. In the case of Arabic, composed rules correctly prefer subject after verb ordering. The composed rules can capture more reordering phenomena because they can contain more terminals and non-terminals, which otherwise would be translated independently by several minimal rules.

DeNeefe et al. [2007] summarize that the minimal and composed rules extracted with the GHKM algorithm (the GHKM rules) can: be phrase pairs with syntactic annotation, encode contextual constraints, have non-contiguous phrases, be purely structural or reorder their children. The authors also categorize the types of GHKM rules in the following way:

- *Non-lexical rules* that have no lexical items on the source side, such as purely structural rules that can potentially be applied to any source sentence.
- *Lexical rules* that have lexical items and outnumber the previous category. Lexical rules are further distinguished in:
  - *Phrasal rules* for which the source side and corresponding target side have one contiguous phrase with any number of variables on either side.

[2] The index of the PP node changes in the resulting composed rule since it remains the only non-terminal.

  - *Non-phrasal rules* that include structural rules, re-ordering rules and non-contiguous phrases.

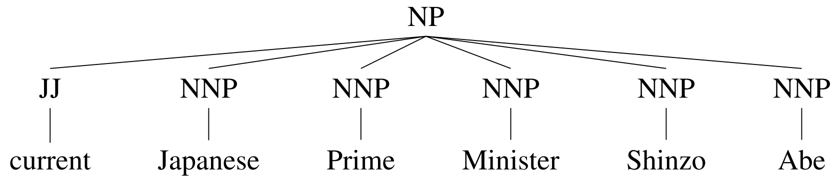
Furthermore, they analyze phrasal coverage by comparing phrasal rules with phrase pairs extracted by a phrase-based system. GHKM rules manage to cover some phrases not extracted by the phrase-based system. This is possible because GHKM rules have no limitation on phrase size. Another more important reason is that unaligned words make it impossible for the phrase-based extractor to extract some phrases while the GHKM extractor can attach the unaligned words at syntactically motivated locations and therefore cover these examples.

Overall the phrase-based system is able to extract many more phrase pairs than the GHKM extractor. DeNeefe et al. [2007] point out that an important deficiency of the GHKM extractor is that it does not learn many of the “useful rules” that the phrase-based decoder uses to build the best translation. One way to recover significantly more phrase-pairs is to increase the *size* of the composed rules. To avoid extracting exponentially many composed rules some limitations are imposed on the size of the rules. The authors define the size of the rule as the number of non-part-of-speech, non-leaf constituent labels in the target tree. The GHKM implementation within the Moses system [Williams and Koehn, 2012] imposes another limitation on the *number of nodes* not counting target words and a limitation on the *rule depth*. The *rule depth* is computed as the maximum distance from the root node to any of its children, not counting pre-terminal nodes. The corresponding Moses parameters are: *MaxRuleSize* for the rule size, *MaxNodes* for the number of nodes and *MaxRuleDepth* for the rule depth. In Chapter 3 we determine what are the optimal values for these parameters for the German→English language pair.

2.2.2 Tree Restructuring

Some phrase-pairs are covered by both phrase-based and string-to-tree translation models. However, the GHKM rules have syntactic constraints for where these phrases can be applied. This can be a strength when the syntactic context can be used to attach the phrase in a syntactically correct way. It can also be a weakness if the syntactic context is too restrictive. Such a case occurs with large, flat, noun phrase (NP) structures as shown in Figure 2.5. While a phrase-based system can apply a phrase-pair to translate only *“Prime Minister”*, a string-to-tree system requires a very specific GHKM rule that covers the entire NP structure. The syntactic constraints of this rule, with several

NNP non-terminals, are too restrictive and cannot be applied in other contexts.



$$NP \rightarrow JJ_0\ NNP_1\ NNP_2\ NNP_3\ NNP_4\ NNP_5\ |||\ X_0\ X_1\ X_2\ X_3\ X_4\ X_5$$

Figure 2.5: Example of a large noun phrase (NP) constituent and the corresponding GHKM rule that would cover this flat structure.

To soften such constraints, and improve grammar coverage, a possible solution is tree restructuring. Tree restructuring strategies such as *binarization* aim to factorize the trees in a way that allows more sub-structures to be extracted, resulting in improved grammar coverage. In this section we describe a few simple binarization strategies: *left binarization*, *right binarization* and *head binarization*. Other more complex binarization strategies have been proposed and described in [Wang et al., 2007].

By *left binarization* all the left-most children of a parent node n, except the right most child, are grouped under a new node. This node is inserted as the left child of n and receives a new label $\bar{n}$. *Left binarization* is then applied recursively to all new nodes until the leaves are reached. *Right binarization*, exemplified in Figure 2.6, implies a similar procedure but in this case the right-most children of the parent node are grouped together except the left most child. *Head binarization* will left-binarize a constituent if the head is the first child and right-binarize it otherwise.

Some constituents of the syntax tree can be factorized only by left-binarization, others only by right-binarization. Therefore, deterministically choosing to apply only one binarization strategy to the entire syntax tree would be a sub-optimal solution. A *parallel binarization* strategy will try to apply both left and right binarization recursively to any parent node with more than two children. This results in a packed binarization forest that allows efficient rule extraction with a dynamic programming algorithm. Forest nodes will encode alternative binarized structures of the original syntax tree nodes. *Parallel head binarization* is a case of *parallel binarization* with the additional constraint that the head constituent is part of all the new nodes created by either left or right binarization.

Because all binarization strategies were shown to bring improvements in BLEU

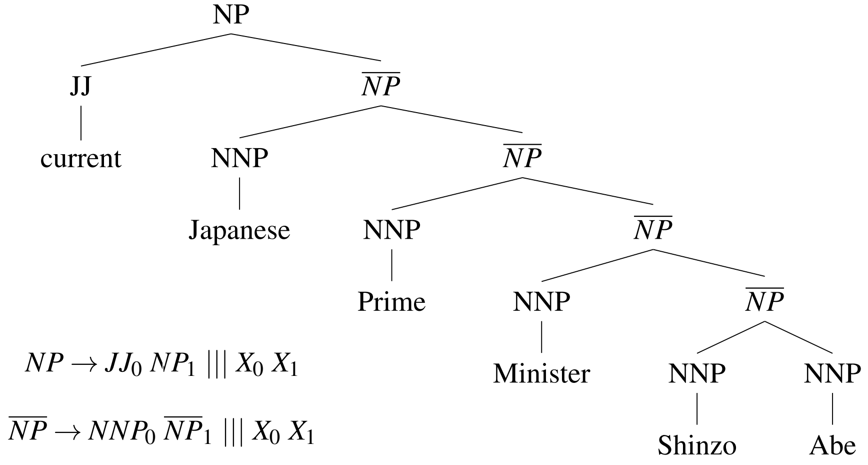


Figure 2.6: Example of a large noun phrase (NP) constituent that is right-binarized and of GHKM rules that can be extracted from this structure.

scores [Wang et al., 2007] and we do not apply the GHKM algorithm to a parse forest, we prefer to use the simpler and more efficient left and right binarization strategies in our work. In Chapter 3 we compare these strategies and corresponding extraction parameters for the German→English language pair.

2.2.3 Incorporating Linguistic Information

String-to-tree systems, with their structured translation rules, allow reordering by abstracting away from the lexical realization of the different syntactic constituents. However, the abstraction reduces the available lexical context and induces translation errors such as incoherent lexical choices and missing words. Furthermore, the size of SCFG rules is controlled to allow for efficient decoding and to avoid problems with data sparsity. This also limits the sentence-level syntactic and lexical context, and induces errors such as incomplete or semantically incoherent predicate-argument structures. To address these issues, previous work has tried to include more sentence-level linguistic information in string-to-tree systems.

Wu and Fung [2009b] showed that semantic roles are preserved across languages in cases where the syntactic roles are not, indicating that more consistent cross-lingual patterns could be induced from shallow semantic frames. Semantic roles are a form of shallow semantics that describe the relation between predicates and their arguments, identifying event structures like "*who* gave *what* to *whom*". Previous work on using semantic role labels (SRL) for SCFG-based SMT has had two main directions: re-

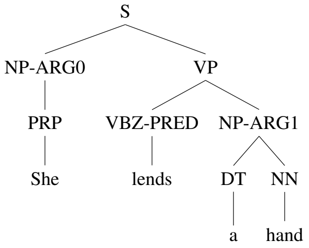


Figure 2.7: Smallest target-side tree fragment that covers all the arguments of the predicate *"lends"* in the example *"She lends a hand"*.

ordering the predicate and its semantic roles [Liu and Gildea, 2010, Li et al., 2013] and extracting rules that cover complete predicate-argument structures [Gao and Vogel, 2011, Bazrafshan and Gildea, 2013].

For string-to-tree systems, Bazrafshan and Gildea [2013] proposed extracting SCFG rules that cover complete predicate-argument structures. The GHKM extraction algorithm was modified such that, for each predicate, a SCFG rule is extracted that has the smallest tree fragment on the target side covering either all or none of the predicate's arguments. Figure 2.7 shows the tree fragment extracted for the verb "lends" in the example "She lends a hand". The semantic role labels of the predicate and its arguments are added to the constituent labels. The rules covering complete semantic frames are added to the original GHKM rules and lead to improved BLEU scores for Chinese→English. Some example translations show better translation of predicates, translation of complete semantic structures and improved ordering of the semantic roles. However, the authors do not mention if they included composed GHKM rules in the grammar of the baseline string-to-tree system. These rules also cover some of the predicate-argument structures and would partially account for the improvements reported. In Chapters 4 and 5, we represent the predicate-argument structure with dependency relations, instead of SRL, since dependency parsers have a higher accuracy and the representation covers the entire sentence. Another difference is that we focus on improving lexical coherence instead of reordering, by modeling semantic affinities between predicates and their argument.

Another approach to integrating more linguistic information in SCFG-based SMT systems is using feature rich discriminative classifiers for rule selection [Braune et al., 2015, 2016, Liu et al., 2008]. Rule selection involves choosing the correct target side of a SCFG rule, considering features of the source side of the rule and surrounding source words. Braune et al. [2015] proposed a discriminative rule selection model for string-to-tree systems using features such as the shape of the source-side of the SCFG rule and the syntactic structure of the source span. However, the authors found that most of the variability in the competing translation options was lexical and not structural. Since their model did not include sentence-level lexical context relevant for lexical disambiguation, translation quality did not improve. In Chapter 5, we propose a verb lexicon model for string-to-tree systems which uses sentence-level lexical context extracted by following the dependency relations of the source verb.

Tamchyna et al. [2016] proposed a discriminative lexicon model for phrase-based SMT integrating complex source and target linguistic feature templates. However, complex target feature templates cannot easily be used in string-to-tree systems because of the hierarchical chart-based decoding strategy[3]. Instead, [Sennrich, 2015] proposed a language model over syntactic n-grams extracted from the target-side of the SCFG rules, using dependency relations as syntactic annotation. In Chapter 4 we compare this model with our proposed Selectional Preferences feature modeling semantic affinities between target-side predicates and their arguments. Further efforts have been dedicated to enriching the lexicon with additional linguistic features. [Williams and Koehn, 2011] augment the string-to-tree system with feature structures encoding morpho-syntactic attributes of target words and enforce unification-based constraints to encourage morphological agreement.

2.3 Neural Language Models

The effectiveness of traditional n-gram language models (LM) is impaired by data sparsity, rigid back-off strategies and lack of generalization. To understand the issue with lack of generalization, consider the following 3-grams: “Eating a banana” and “Eating an apple”. The LM sees these n-grams as distinct inputs and does not model the semantic or syntactic similarity between the two objects: “banana” and “apple”.

Depending on the domain and amount of training data, semantically similar n-grams, for example “Eating a Kiwano”, might not be seen at all during training. In such cases different backing-off strategies are applied, such as interpolating the probabilities of the bi-grams “Eating a” and “a Kiwano”. However, these strategies are restrictive

[3] Integrating a language model is also challenging in chart-based decoders, since the target context of the lower chart cells has to be remembered until a SCFG rule is applied at later stages, combining and possibly re-ordering the content of these cells. In phrase-based SMT, decoding is sequential left-to-right and not hierarchical, and the target context of the previous phrase-pair is discarded as soon as the next phrase-pair is scored.

and cannot be adapted according to context. Data sparsity is an issue for n-gram LMs, as the number of free parameters grows exponentially with n: $|V|^n$, where $|V|$ is the size of the vocabulary.

Neural language models address these issues by representing words in a continuous space where similar words are close to each other. The probability of a word sequence is computed based on these distributed representations, which are combined using highly parameterized non-linear functions. Because the resulting probability function is smooth, substituting the word "banana" with a similar word such as "apple", will result in only a small change in the probability of the word sequence [Bengio et al., 2003]. Furthermore, neural LMs can be extended with source context and used for generation, which forms the basis of neural MT.

In this section we describe two main types of neural LMs, one based on feed forward neural networks and the other on recurrent neural networks.

2.3.1 Feed Forward Neural Networks

Similar to n-gram LMs, the neural probabilistic language model (NPLM) [Bengio et al., 2003] is a function, implemented by a feed forward neural network, estimating the probability of the next word w_k conditioned on the previous $n-1$ words: $p(w_k|w_{k-n+1},...,w_{k-1})$. Next, we formalize the three components of an NPLM: *embedding layer*, *hidden layer* and *softmax layer*.

The *embedding layer* maps each word w_k from the vocabulary V to its continuous-space representation, an m-dimensional feature vector $e_k \in \mathbb{R}^m$, where m is much smaller than $|V|$. The free parameters that have to be learned for the embedding layer are represented by a matrix $E \in \mathbb{R}^{|V| \times m}$, where each row of the matrix corresponds to a word embedding. The representation of the conditioning context is obtained by concatenating the $n-1$ word feature vectors: $c = [e_{k-n+1};..;e_{k-1}]^\top$, with $c \in \mathbb{R}^{m*(n-1)}$.

The *hidden layer* takes as input the context vector c, applies an affine transformation followed by a non-linear function ϕ and outputs a vector h_1 representing higher-order features of the input sequence.

$$h_1 = \phi(Hc + d) \tag{2.11}$$

where $H \in \mathbb{R}^{h \times m*(n-1)}$, $h_1, d \in \mathbb{R}^h$ and ϕ is the activation function. h is the number of hidden units, m the number of word features, $n-1$ the size of the context.

Several such layers can be composed in order to learn more abstract features. Some of the activation functions that have been reported in the literature are the *tanh* ($\phi(x) = \frac{1-exp(-2x)}{1+exp(-2x)}$), the sigmoid ($\phi(x) = \frac{1}{1+exp(-x)}$) and the *rectified linear unit (RelU)*($\phi(x) = \max(0,x)$).

The output of the last hidden layer is used to compute the probability distribution over the target word vocabulary, with the help of a *softmax* layer. This layer applies an affine transformation to the hidden state h_1, to obtain an unnormalized score for each target word w_k and then normalizes these scores to obtain a probability distribution $\hat{P}$.

$$y = Uh_1 + b \tag{2.12}$$

$$\hat{P}(w_k|w_{k-n+1},...,w_{k-1}) = \frac{exp(y_{w_k})}{\sum_{i=1}^{|V|} exp(y_i)} \tag{2.13}$$

where $U \in \mathbb{R}^{|V|\times h}$, $y, b \in \mathbb{R}^{|V|}$ and y_{w_k} is the unnormalized score for the word w_k.

The parameters $\theta = (d,b,H,U,E)$ of this feed forward neural network with one hidden layer are learned jointly to maximize the log-likelihood of the training data. The optimization is performed with back propagation and stochastic gradient ascent[4]. The total number of parameters[5] scales linearly with the size of the context n and with the size of the vocabulary $|V|$, compared to the traditional n-gram LMs where the number of parameters grows exponentially $|V|^n$.

One important computational bottleneck of NPLMs is that most of the computation happens in the output layer which can account for more than 99% of the computation[6] [Bengio et al., 2003]. One solution to this problem is to train self-normalizing models, such that at inference time there is no need to compute the normalization factor [Vaswani et al., 2013, Devlin et al., 2014]. Several NPLMs using some type of self-normalization or other optimization tricks have been used successfully as an extra feature in the decoder of a phrase-based SMT model [Schwenk et al., 2006, Vaswani et al., 2013, Devlin et al., 2014, Baltescu et al., 2014].

Still, using an NPLM as a feature for SMT systems is efficient only when n is small which does not allow for modeling long-distance dependencies. Some have incorporated additional context by looking at the words in the source sentence [Devlin et al., 2014, Schwenk, 2012]. In Chapter 5 we use a similar approach to incorporate global source-side syntactic information for a Neural Verb Lexicon Model. Another

[4] An alternative formulation is to minimize negative log-likelihood with stochastic gradient descent.

[5] $\sum dim(d,b,H,U,E) = h + |V| + h*m*(n-1) + |V|*h + |V|*m = |V|*(1+h) + h*(m*(n-1)+1)$

[6] Considering $|V| = 500{,}000$, $h = 512$, $m = 100$, n=15 then the fraction of the total operations can be approximated as $(500{,}000*513)/(500{,}000*513+513*(100*14+1)) = 0.997$

way to incorporate some long-distance dependencies is to condition the model on syntactic n-grams which are extracted from the target-side of a string-to-tree SMT system [Sennrich, 2015]. In Chapter 4 we compare this model with our proposed Selectional Preferences feature modeling semantic affinities between target-side predicates and their arguments.

2.3.2 Recurrent Neural Networks

NPLMs are able to generalize better than traditional n-gram LMs and also to model larger contexts with fewer parameters. However, the input to the network has a fixed dimensionality and is limited in practice because the model size scales linearly with the size of the context. In contrast, recurrent neural networks (RNNs) are able to incorporate the entire, variable-length word history as context and implicitly learn long-distance dependencies.

An RNN LM [Mikolov et al., 2010] summarizes a variable-length word sequence $w_1, ..., w_{t-1}$ by reading one input word at each time step $t' \in [1, t-1]$ and updating the hidden state h_t, a fixed-dimensional vector representation, using a recurrent function whose parameters are shared across time steps. The last hidden state h_{t-1} is then transformed using a *softmax* layer into the probability distribution of the next word w_t.

$$h_t = \begin{cases} \phi(W_{wh}e_{w_t} + W_{hh}h_{t-1} + b_h), & \text{if } t >= 1 \\ 0, & \text{if } t = 0 \end{cases} \tag{2.14}$$

$$P(w_t | w_1, ..., w_{t-1}) = softmax(h_{t-1}) \tag{2.15}$$

$$P(w) = \prod_{t=1}^{T} P(w_t | w_1, ..., w_{t-1}) \tag{2.16}$$

In principle, the hidden state h_t is able to memorize information from the beginning of the sequence and pass it on to later time steps. On the other hand, to update the parameters, the loss corresponding to predictions made at later time steps has to back-propagate through time up to the first state. Because of the *vanishing gradient* problem the updates to the parameters might become insignificant and the model is not trained adequately.

The *long short-term memory (LSTM)* neural network [Hochreiter and Schmidhuber, 1997] avoids the vanishing gradient problem by introducing a *memory cell*[7]. The

[7] The derivative of the memory cell $\frac{\partial c_t}{\partial c_{t-1}}$ is close to 1.

LSTM also introduces a gating mechanism which controls the amount of information from the current input that can update the memory cell and the amount of information from the memory cell that is passed as the current hidden state. First, the previous hidden state is partially updated given the current input to get an intermediate state u. This intermediate state is modulated by the input gate i to allow some of the hidden units to be added to the previous memory cell. To obtain the new hidden state the updated memory cell is transformed using a non-linear function and the result is modulated by the output gate o.

$$u_t = \tanh(W_{xu}x_t + W_{hu}h_{t-1} + b_u) \tag{2.17}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{2.18}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{2.19}$$

$$c_t = i_t \odot u_t + c_{t-1} \tag{2.20}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.21}$$

Several extensions to LSTMs have been proposed such as adding a *forget gate* which resets part of the memory cell or simplifying the equations as in the *gated recurrent unit* (GRU)[Cho et al., 2014b].

The memory cell and the gating mechanisms allow LSTMs and GRUs to summarize long sequences and to capture some long distance dependencies. These two properties turn out to be essential for modeling the translation of an input sequence into a target sequence. Moreover, a new sequence can be generated with RNNs by iteratively sampling from the distribution over the next word $P(w_t|w_1,...,w_{t-1})$. To generate a translation, an input sentence can be summarized using one RNN and the last hidden state can be passed as context for a second RNN which outputs one target word at a time. In the next section we see how this simple idea can be improved to build a state-of-the-art NMT system.

2.4 End-to-end Neural Machine Translation

2.4.1 Encoder-decoder RNN

To learn a neural translation model, Sutskever et al. [2014], Cho et al. [2014b] proposed encoding the source sentence x using an RNN and then using the last hidden state as context for another RNN which generates a target sentence y. The $encoder-decoder$ RNN learns the conditional probability distribution of the sequence of target words given the sequence of source words, which is represented by the last hidden state c of the encoder RNN:

$$P(y_1,....y_T|x_1,...x_S) = \prod_{t=1}^{T} P(y_t|c,y_1,...y_{t-1}) \tag{2.22}$$

$$P(y_t|c,y_1,...y_{t-1}) = g(c,h_t,y_{t-1}) \tag{2.23}$$

$$h_t = f(h_{t-1},y_{t-1},c) \tag{2.24}$$

where g is the softmax function which outputs a probability distribution over the target vocabulary, and f is an RNN function such as the LSTM in Sutskever et al. [2014] or the GRU in Cho et al. [2014b]. Kalchbrenner and Blunsom [2013] proposed to use a convolutional neural network for the encoder, which to some extent could implicitly learn the hierarchical aspects of the source language without using an explicit syntactic representation.

The $encoder-decoder$ RNN showed promising results as an end-to-end machine translation model [Sutskever et al., 2014] and was also used to improve the output of an SMT system by rescoring an n-best list [Cho et al., 2014b].

2.4.2 Encoder-decoder RNN with Attention

A major drawback of the encoder-decoder RNN is that performance degrades significantly as the length of the sentences increases [Cho et al., 2014a]. Sutskever et al. [2014] claim that by reversing the order of the source words they create more short-term dependencies between the first target words and their corresponding source words, thus solving the issues with back-propagation for long sequences[8]. This might be true for mostly monotonic translation, as is the case for the French→English language pair which was used in their evaluation. However, in the case of target languages which have flexible word order, such as German, it does not seem possible to create short-term dependencies just by reversing the order of the source sentence.

Bahdanau et al. [2015] argue that the $encoder-decoder$ RNN cannot handle long sentences because the fixed-size context vector does not have enough capacity to sum-

[8] The problem of learning the model parameters with back-propagation for long sequences is exacerbated in the translation scenario since there are approximately twice as many time-steps than in a monolingual scenario. The partial derivatives are computed and multiplied for all the source and target RNN states.

marize a long sentence[9]. Instead, they propose to adapt the source context for each target word by computing a soft-attention over all encoder states. The adapted context c_i is a weighted sum over the encoder states, where the weights quantify how relevant is a particular source state s_j for computing the current target state h_i. The attention weights α_{ij} are recomputed for each target word, based on the activations e_{ij} of a feed-forward neural network a which is conditioned on the previous decoder state h_{i-1} and the encoder state s_j. All the components of the network, including the attention model a, are trained jointly similar to an $encoder - decoder$ RNN. Another extension proposed by Bahdanau et al. [2015] is to use a bi-RNN encoder which concatenates the states $\overrightarrow{s_j}$ of a forward-pass RNN with the states $\overleftarrow{s_j}$ of a backward-pass RNN. This new hidden state $s_j = [\overrightarrow{s_j}; \overleftarrow{s_j}]$ summarizes both the preceding and following words, and is not biased by more recent inputs as was the case in the simple RNNs. We formalize below the new attention model, decoder and conditional probability distribution of a target sequence.

$$s_j = [\overrightarrow{s_j}; \overleftarrow{s_j}] \tag{2.25}$$

$$e_{ij} = a(h_{i-1}, s_j) \tag{2.26}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{S} \exp(e_{ik})} \tag{2.27}$$

$$c_i = \sum_{j=1}^{S} \alpha_{ij} s_j \tag{2.28}$$

$$h_i = f(c_i, h_{i-1}, y_{i-1}) \tag{2.29}$$

$$P(y_i | c_i, y_1, ...y_{i-1}) = g(c_i, h_i, y_{i-1}) \tag{2.30}$$

$$P(y_1, ...y_T | x_1, ...x_S) = \prod_{i=1}^{T} P(y_i | c_i, y_1, ...y_{i-1}) \tag{2.31}$$

where g is the softmax function which outputs a probability distribution over the target vocabulary, and f is an RNN function such as the LSTM or GRU.

Bahdanau et al. [2015] report significant improvements with this new architecture over the baseline $encoder - decoder$ RNN, in particular for long sentences. However, they do not present ablation results to determine how much of the improvement is due to the bi-RNN and how much due to the attention model. The authors also point out that

[9] If the size of the hidden state is increased, then all the other parameter matrices have to be scaled accordingly. This results in a larger number of parameters which will be harder to learn given limited training data.

the limited target vocabulary significantly affects the performance of the NMT system, since generating UNK tokens (for unknown words) results in lower BLEU scores.

To allow for open source and target vocabularies, Sennrich et al. [2016b] propose splitting words into byte-pair-encoding (BPE) sub-units. The BPE sub-units are learned by iteratively merging the most frequent character sequences, starting with single characters and stopping when the maximum number of allowed merge operations has been reached. The resulting vocabulary size is equal to the number of initial characters plus the number of allowed merges. Since the most frequent words will be represented by one BPE unit, this encoding offers a good trade-off between the size of the target vocabulary and the length of the encoded sequence. This trade-off is important because the time complexity of NMT is linear in the size of the target vocabulary and super linear in the lengths of the source and target sequences. Other approaches to open vocabulary include using character-level sequences [Chung et al., 2016] and using Huffman encoding [Chitnis and DeNero, 2015].

In Chapter 6 we use the Nematus toolkit [Sennrich et al., 2017] implementing an encoder-decoder RNN with attention and computing the f function using two GRU layers coupled by the attention model. We also use BPE encoding to represent source and target words.

2.4.3 Strengths and Limitations of NMT

Bentivogli et al. [2016] perform a detailed analysis of the NMT and phrase-based SMT output for English→German, showing that NMT improves translation quality in particular with respect to morphology and word order. Aspects that can still be improved include modeling of long sentences and reordering of prepositional phrases (PP) and subjects.

Sennrich [2017] evaluate the performance of NMT systems for English→German on several linguistic phenomena using contrastive translations with automatically induced errors. The evaluation is in terms of the accuracy with which the system distinguishes the reference translation from the contrastive translation. The results show accuracy is about 95% for subject-verb agreement of adjacent words, but goes down to 90% when the distance between the two words increases. Other types of errors that are still problematic for NMT, with accuracies around 90%, are the deletion of negative polarity markers and translation of separable verb particles[10].

[10] Some German verbs have particles which are separated from the main verb and often placed at the end of the sentence.

Shi et al. [2016] study the extent to which the NMT encoder learns syntactic information about the source language. They use the hidden states of a pre-trained NMT encoder to predict syntactic labels such as voice and tense, or the entire serialized parse tree of the source sentence. They find that the NMT encoder captures more sentence-level syntactic information compared to an auto-encoder. However, the NMT encoder induces roughly twice as many bracketing errors, compared to an encoder trained specifically to predict parse trees[11]. The NMT encoder also confuses the part-of-speech of the head words 16 times more often. Furthermore, prepositional phrase attachment errors are the most prevalent when either encoder is considered.

In a monolingual study, Linzen et al. [2016] show that LSTMs learn subject-verb number agreement with high accuracy. However, when provided with explicit supervision the network is able to learn the agreement more accurately than when it is trained as a language model. Still, more errors occur for complex constructs, when the subject and verb are separated by an interleaving relative clause or multiple attractor nouns with a different number than the subject.

These studies suggest that NMT systems handle some difficult linguistic structures surprisingly well. However, NMT can still be improved with explicit linguistic supervision, in particular on difficult linguistic constructs such as prepositional phrase attachment and on long sentences which involve long distance dependencies.

2.4.4 Incorporating Linguistic Information

Incorporating source-side linguistic information in NMT, either as distant supervision [Luong et al., 2016] or as explicit features in the encoder [Sennrich and Haddow, 2016, Eriguchi et al., 2016], has been previously explored.

Luong et al. [2016] use a multitask learning framework with a shared encoder to co-train a translation model and a source-side syntactic parser. In Chapter 6 we use multitasking to incorporate target-side syntactic information.

Eriguchi et al. [2016] extend the LSTM encoder representing source words in context, with a tree-LSTM encoding the phrase structure of the source sentence. The tree-LSTM constructs the sentence representation by combining recursively and bottom-up the representations of binary phrase structures. The phrase structures are obtained by parsing the source sentence using a head-driven phrase structure grammar (HPSG).

[11] An NMT encoder is coupled with a decoder that predicts the serialized parse tree of the source sentence. The decoder is trained to predict parse trees, without updating the parameters of the encoder. This is compared to an encoder-decoder system trained end-to-end to predict the serialized parse tree.

Sennrich and Haddow [2016] generalize the embedding layer of NMT to include explicit linguistic features such as dependency relations and part-of-speech tags:

$$h_t = RNN(h_{t-1}, [x_t; m_t^1 ...; m_t^K]) \tag{2.32}$$

where there are K features and m_t^k are feature embeddings which are learned jointly with the word embeddings x_t and the other parameters of the NMT model. The source-side linguistic features improve a baseline NMT system, for example by helping with word sense disambiguation: the word "close" will be translated to German as "nah" when used as an adjective and as "schließen" when used as a verb. In Chapter 6 we use this framework to show source and target syntax provide complementary information.

Integrating explicit linguistic information in the decoder is more challenging as this requires incrementally generating words and appropriate linguistic factors with limited target context. Martínez et al. [2016] propose a factored NMT decoder which generates lemmas and morphological tags, with the purpose of reducing the size of the target vocabulary. The factors are generated independently with distinct softmax functions, their corresponding embeddings are combined (e.g. by addition) and the result is used as context for computing the next decoder state. The authors also explore generating the morphological tag conditioned on the lemma embedding. However, neither of the factored decoder architectures lead to an improvement in translation quality. In Chapter 6 we propose a method for incorporating target syntax in the decoder by interleaving the words with their corresponding CCG supertags. Our approach allows for target words to be generated conditioned on their syntactic category and the previous lexical and syntactic context.

2.5 Syntactic Representations

In this section we describe the syntactic representations used in this thesis to extract sentence level information, such as the subcategorization frame of verbs.

2.5.1 Dependency relations

Dependency relations represent the grammatical structure of a sentence as binary relations between a *head* and a *dependent*, which cover the entire sentence forming either a tree or a graph. The Stanford dependencies manual [de Marneffe and Manning, 2008] defines a set of 50 grammatical relation types for English and a *basic representation* in which each word in the sentence participates in a relation, such that a dependency *tree* is formed.

The *collapsed representation* of Stanford dependencies merges dependencies involving function words, such as prepositions and conjuncts, in order to obtain direct dependencies between content words. This representation is suitable for representing the syntactic-semantic relations between verbal or nominal predicates and their arguments. We give an example of the two representations in Figure 2.8.

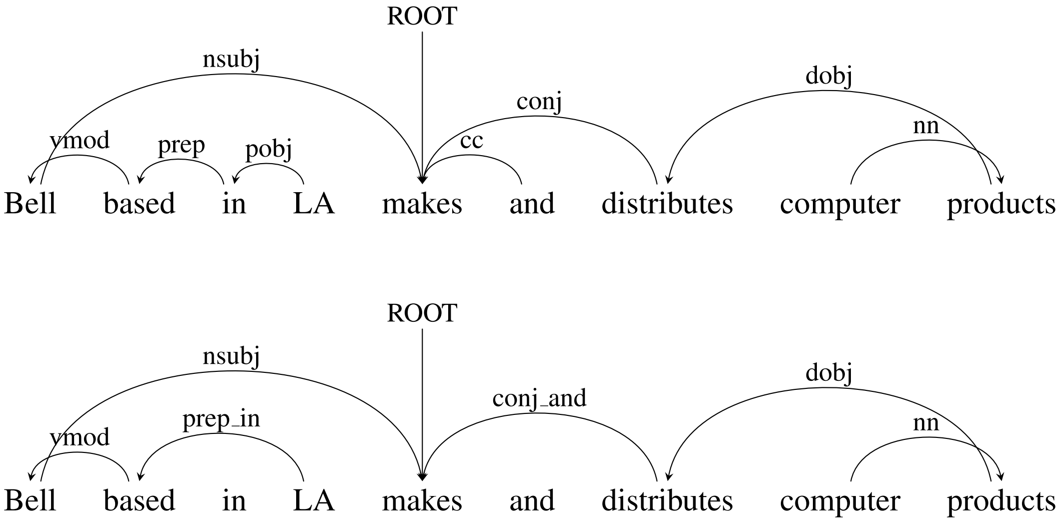

ROOT
nsubj
conj
dobj
nn
vmod
prep
pobj
cc
Bell based in LA makes and distributes computer products
ROOT
nsubj
conj_and
dobj
nn
vmod
prep_in
Bell based in LA makes and distributes computer products

Figure 2.8: Example of Stanford typed dependency relations for the sentence *"Bell, based in LA, makes and distributes computer products."*. The *basic representation* is shown on top and the *collapsed representation* on the bottom (adapted from de Marneffe and Manning [2008]).

We use the Stanford typed dependencies in Chapter 4 to model selectional preferences of verbs and nouns in string-to-tree systems. For prepositional modifiers we use the collapsed representation, which we obtain by processing the *prep* and *pobj* tree nodes during decoding. The GHKM rule extraction algorithm used for training a dependency-based string-to-tree system requires dependencies to be converted from the *basic representation* to constituency representation.

A dependency representation can be converted into a constituency tree by first applying heuristic projectivization [Nivre and Nilsson, 2005] (resulting in a projective dependency graph) followed by a lossless conversion [Sennrich and Haddow, 2015]. A dependency graph is projective if all its arcs are projective. An arc is projective if all the words inside its span are connected to words within this span. The algorithm for graph projectivization [Nivre and Nilsson, 2005] applies lifting operations on non-projective arcs to connect the dependent to the head word of its original head. The constituency representation allows only one pre-terminal child for each node, which is the head word of that span.

Figure 2.9 gives an example of a phrase annotated with the Stanford Neural Network dependency parser [Chen and Manning, 2014b] (left), and its constituency representation (right). The node *NNP* is the only pre-terminal child of the *ROOT* node and the corresponding terminal node *"Minister"* is the head word of the entire span.

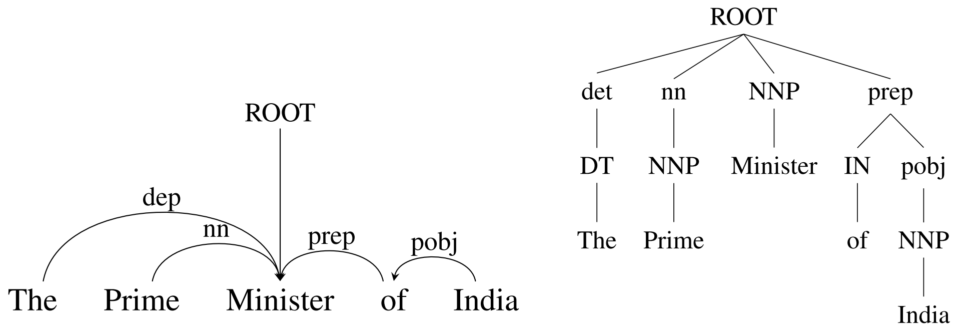


Figure 2.9: Dependency relations (left) and the corresponding constituency tree (right).

Different dependency relations are defined for languages other than English. In Chapter 6 we use the ParZU [Sennrich et al., 2013] dependency parser for German which is trained on the TüBa-D/Z dependency tree bank Gastel et al. [2011] and the SyntaxNet [Andor et al., 2016] dependency parser for Romanian trained on the Universal Dependencies (UD) treebank [Nivre et al., 2016].

The UD proposes a reduced set of 40 grammatical relations that allow consistent annotation across languages. We list the UD dependency relations in Table 2.2. The ParZU parser assigns the special label *avz* to separable verb particles, which are placed at the end of the sentence[12]. We use this dependency relation to provide relevant source-side context for verb translation in Chapter 5.

2.5.2 CCG supertags

Combinatory categorial grammar (CCG) is a lexicalized formalism in which words are assigned with syntactic categories, i.e., *supertags*, that indicate context-sensitive morpho-syntactic properties of a word in a sentence. The CCG supertag can be either

[12] For example, the verb particle *an* in the sentence *Er kommt morgen an.*.

Core dependents of clausal predicates		
Nominal dependency	*Predicate dependency*	*Other*
nsubj	csubj	xcomp
nsubjpass	csubjpass	
dobj	ccomp	
iobj		
Non-core dependents of clausal predicates		
Nominal dependency	*Predicate dependency*	*Modifier word*
nmod	advcl	advmod
		neg
Special clausal dependents		
Nominal dependency	*Auxiliary*	*Other*
vocative	aux	mark
discourse	auxpass	punct
expl	cop	
Noun dependents		
Nominal dependency	*Auxiliary*	*Other*
nummod	acl	amod
appos		det
nmod		neg
Case-marking, prepositions, possessive		
case		
Coordination		
conj	cc	punct
Compounding and unanalyzed		
compound	mwe	goeswith
name	foreign	
Loose joining relations		
list	parataxis	remnant
dislocated	reparandum	
Other		
Sentence head	*Unspecified dependency*	
root	dep	

Table 2.2: The Universal Dependencies (UD) relations. Reproduced from Nivre et al. [2016].

a primitive type or a complex type, which is interpreted as a function accepting arguments and returning a result. For a complex type, the CCG supertag describes the type of its arguments, the order in which these are accepted and the type of the result.

A complex type has the form X/Y or $X\backslash Y$, where X and Y can be a primitive type, such as *S, NP, VP*, or a complex type. The rightward-combining functor X/Y accepts an argument of type Y to the right and returns a result of type X. The leftward-combining functor $X\backslash Y$ accepts an argument of type Y to the left and returns a result of type X.

Figure 6.2 gives an example of a sentence annotated with CCG supertags. The supertag for the ditransitive verb *"receives"*, $((\mathsf{S[dcl]}\backslash\mathsf{NP})/\mathsf{PP})/\mathsf{NP}$, encodes the subcategorization frame of the verb: two arguments to the right, the first being a prepositional phrase (PP) and the second a noun phrase (NP), and one NP argument (subject) to the left. After the verb accepts the rightward arguments, the result is another function that maps a subject NP into a sentence (S). The CCG supertag also exhibits the feature *[dcl]*, indicating that the resulting sentence is declarative. Features can encode, for example, whether a sentence is a question or the tense of the verb.

CCG defines combinators which allow the composition of adjacent supertags. The successive application of combinators, represented with a derivation tree, results in a final category, usually of type *S*. The combinators include forward and backward application, forward and backward composition and type raising.

Several CCG supertags can be assigned to the same word, depending on its context. A CCG supertagger ranks each possible supertag, based on sentence-level features. Given the possible CCG supertags for each word, a CCG parser finds the most probable sequence which forms a valid derivation. In our work, we consider only the most probable sequence of CCG categories, which we obtain using the EasySRL parser [Lewis et al., 2015], and we discard the derivation tree. We use the most probable CCG supertag sequence to encode sentence level syntactic constrains locally, at word level, which can be easily integrated either in the NMT encoder or decoder.

Consider a decoder that has to generate the following sentences:

1. What city is the Taj Mahal in?
$(S[wq]/(S[q]/NP))/N$ N $(S[q]/PP)/NP$ NP PP/NP

2. Where is the Taj Mahal?
$S[wq]/(S[q]/NP)$ $(S[q]/NP)/NP$ NP

If the decoding starts with predicting "*What*", it is ungrammatical to omit the preposition "*in*", and if the decoding starts with predicting "*Where*", it is ungrammat-

Tokens:	Obama	receives	Netanyahu	in	the	capital	of	USA
CCG:	NP	((S[dcl]\NP)/PP)/NP	NP	PP/NP	NP/N	N	(NP\NP)/NP	NP

Figure 2.10: Example of a CCG supertag annotation.

ical to predict the preposition. Here the decision to predict "*in*" depends on the first word, with several other words in between. However, if we rely on CCG supertags, the supertags of both these sequences look very different. The supertag (S[q]/PP)/NP for the verb "*is*" in the first sentence indicates that a preposition is expected in future context. Furthermore, it is likely to see this particular supertag of the verb in the context of (S[wq]/(S[q]/NP))/N but it is unlikely in the context of S[wq]/(S[q]/NP). Therefore, a succession of local decisions based on CCG supertags will result in the correct prediction of the preposition in the first sentence, and omitting the preposition in the second sentence. Since the vocabulary of CCG supertags is considerably smaller than that of possible words[13], reducing data sparsity, the NMT model is better at generalizing over and predicting the correct CCG supertags sequence.

Predicting the CCG supertag for a target verb can help the NMT decoder generate the correct number of arguments, in the correct order, especially if the source and target languages have different word orders. For the German→English language pair, knowing wether a subordinate clause is expected can also be useful, as this triggers a reordering of the verb and its arguments in English. Figure 2.11 gives an example of a subordinate clause where the German verb *"bezeichnete"* comes at the end, while the corresponding English verb *"referred"* is reordered before the object *"to Prentiss"*. In Chapter 6, we come back to this example showing how an NMT system using CCG supertags in the decoder can correctly handle the re-ordering.

Source	... dass Lamb in seinem Notruf Prentiss zwar als seine Frau bezeichnete ...
Gloss	... that while Lamb in his 911 call Prentiss as his wife referred ...
Reference	... that while Lamb referred to Prentiss as his wife in the 911 call ...

Figure 2.11: Example of re-ordering inside a subordinate clause in German→English translation.

CCG supertags also help during encoding if they are given in the input, as we

[13] In our experiments, hundreds vs ten-thousands.

can see with the case of PP attachment in Figure 2.10. This sentence contains two PP attachments and could lead to several disambiguation possibilities (*"in"* can attach to *"Netanyahu"* or *"receives"*, and *"of"* can attach to *"capital"*, *"Netanyahu"* or *"receives"*). These alternatives may lead to different translations in other languages. However the supertag $((\mathsf{S[dcl]}\backslash\mathsf{NP})/\mathsf{PP})/\mathsf{NP}$ of *"receives"* indicates that the preposition *"in"* attaches to the verb, and the supertag $(\mathsf{NP}\backslash\mathsf{NP})/\mathsf{NP}$ of *"of"* indicates that it attaches to *"capital"*, thereby resolving the ambiguity.

Translation of the correct verb form and agreement can be improved with CCG since supertags also encode tense, morphology and agreements. For example, in the sentence "*It is going to rain*", the supertag $(\mathsf{S[ng]}\backslash\mathsf{NP[expl]})/(\mathsf{S[to]}\backslash\mathsf{NP})$ of "*going*" indicates the current word is a verb in continuous form looking for an infinitive construction on the right, and an expletive pronoun on the left.

In Chapter 6, we show that using CCG supertags as target syntax in a NMT decoder can improve translation quality. We also show that CCG supertags are useful as additional source-side linguistic features in the encoder.

2.6 Conclusion

In this chapter, we presented an overview of the two translation systems used throughout this thesis: string-to-tree SMT and sequence-to-sequence with attention NMT. Both systems are able to model syntax and capture long distance dependencies to some extent. This chapter also described representations for additional linguistic information useful in addressing the limitations of the systems. The remainder of the thesis discusses approaches to incorporate these representations into the systems.

Chapter 3

Improving Robustness of String-to-tree Systems

3.1 Introduction

Training string-to-tree MT systems requires a pipeline of tools for data pre-processing and language annotation. The interaction between these annotation tools greatly impacts the quality and robustness of the resulting SCFG. For example, the extraction of translation rules is affected by the characteristics of the syntactic annotation, such as tree depth and branching direction, as well as by the syntactic constituents that break the alignment heuristics.

In this chapter we explore several methods of improving the robustness of string-to-tree systems for translating into English. Several experiments were conducted for German→English, which is the primary language pair used throughout this thesis. This language pair is challenging for SMT because of the difference in word order and morphological richness between the two languages.

The main contributions we bring at this stage are threefold. Firstly, we propose in Section 3.2 two methods for improving the consistency between the string-to-tree system and the target syntactic representation for German→English translations. More specifically, through the 1) use of an appropriate tokenization strategy and 2) the selection of extraction parameters that match the degree of nestedness of the English syntactic structures Secondly, we propose three methods to make the string-to-tree systems more robust: tree restructuring, realigning verbs and pre-processing named entities. Thirdly, we present experiments with neural language models aimed at improving generalization. These experiments are reported in Section 3.3.

We conclude this chapter with Section 3.4, an error analysis of a string-to-tree system for German→English, which highlights aspects that can be improved using semantic and syntactic features. The string-to-tree systems described in this chapter were state-of-the-art for German→English at WMT2013 and WMT2014, mainly due to improving the robustness of the SCFG. In Table 3.1 we summarize the progress of the string-to-tree systems for German→English over several WMT evaluation campaigns. We report both BLEU scores and the ranking according to manual evaluation. We compare the results with those of a phrase-based system [Durrani et al., 2013, 2014, Haddow et al., 2015, Williams et al., 2016] and of a neural machine translation system [Jean et al., 2015, Sennrich et al., 2016a]. The referenced syntax-based systems were trained by the author of this thesis. The novelties of each system are described in this chapter.

	2013		2014		2015		2016	
System	BLEU	Rank	BLEU	Rank	BLEU	Rank	BLEU	Rank
Phrase-based	**26.6**	4-5	28.0	4-6	**29.3**	**2-3**	35.1	5-7
Syntax-based	26.3	**2-3**	**28.2**	**2-3**	28.7	3-5	34.4	2-5
Neural	-	-	-	-	27.6	6-7	**38.6**	**1**

Table 3.1: Comparing state-of-the-art translation systems for German → English that participated at WMT2013-2016. We report cased BLEU scores as well as the ranking interval for each system according to manual evaluation.

For German→English, the referenced phrase-based systems perform word reordering as a pre-processing step using the clause restructuring method proposed by Collins et al. [2005]. This method relies on the parse tree of the source sentence and on linguistically-motivated transformations to reorder the verb according to the English word order. As discussed in Section 2.5, in German, the verb is placed at the end of the subordinate clause. One of the transformations proposed by Collins et al. [2005] accounts for reordering the verb in a subordinate clause. The proposed transformations also include moving the subject to directly precede the head of the clause and moving the separable verb particle to immediately precede the verb.

At WMT2013 and WMT2014, the string-to-tree system was ranked highest among the constrained systems, with the first rank going to online commercial systems trained on larger data sets. Indirectly, this shows that the string-to-tree systems perform better at long-distance reordering of the verb and its arguments than the pre-reordering method proposed for phrase-based systems.

At WMT2016, a neural machine translation system performed much better than both the string-to-tree and phrase-based systems. These results underlie our decision to explore the benefit of integrating target-syntax in a neural machine translation system, which we describe in Chapter 6.

3.2 Baseline setup

In this section we describe a baseline string-to-tree system for translating into English. We introduce two methods for making the string-to-tree systems more consistent with the target syntactic representation: using an appropriate tokenization strategy and selecting extraction parameters that match the degree of nestedness of the English syntactic structures. The experiments reported in this section were previously described in Nădejde et al. [2013].

The string-to-tree system used across all experiments is trained with the Moses toolkit implementing GHKM rule extraction and Scope-3 parsing [Williams and Koehn, 2012]. The English side of the parallel corpus was parsed using the Berkeley parser [Petrov et al., 2006] and German compounds were split using the script provided with Moses. The parallel corpus was word-aligned using MGIZA++ [Gao and Vogel, 2008]. 5-gram language models were trained using SRILM toolkit [Stolcke, 2002] with modified Kneser-Ney smoothing [Chen and Goodman, 1998] and then interpolated using weights tuned on the development set. The feature weights for each system were tuned on development sets using the Moses implementation of minimum error rate training [Och, 2003].

The datasets used throughout this thesis were provided by the WMT evaluation campaigns [Federmann et al., 2013, Bojar et al., 2014, 2015, 2016]. In Table 3.2 we give the number of parallel sentences used for training, tuning and evaluating the baseline German→English string-to-tree system. We also use all available monolingual data for the language models. The systems described in the following sections were trained on data provided at different editions of WMT. This allowed us to compare results with the other state-of-the-art systems on the same evaluation sets. However, for German→English, the data sets had similar sizes for all experiments, and we report the corpus statistics in Appendix A. Throughout this thesis, we use cased BLEU [Papineni et al., 2002] as the automatic evaluation metric, which computes the modified (by clipping) corpus-level precision over n-grams found in the machine translation, compared to those in the reference translation.

Language pair	Train	Tune	Test
German→English	4,434,060	2,400	3,000

Table 3.2: Corpus statistics for parallel data. The number of sentences is reported for the WMT13 datasets.

Tokenization We make the rule-extraction process more consistent with the target syntactic representation by matching the tokenization strategy for English sentences with that of the syntactic parser. This results in fewer parser errors affecting the quality of the extracted synchronous grammar. We change the quotation marks, which are very frequent in the training data, to opening and closing quotation marks to match the punctuation style of the Penn Treebank. We also added Penn Treebank style tokenization rules[1]. These rules split contractions such as *I'll* → *I 'll*, *Don't* → *Do n't*, *Maria's* → *Maria 's*, in order to correctly separate the verbs, negation and possessives that are parsed as separate constituents. By dealing with these contractions, word alignment becomes more consistent and more synchronous rules can be extracted.

Rule Extraction We performed experiments for the German→English language pair to determine the optimal parameters for the rule extraction algorithm, which we described in Section 2.2.1.2: *Rule Depth*, *Rule Size*, *Node Count*. For efficiency reasons we used a subset of the parallel training data. We chose the parameters
Rule Depth=5, Node Count=20, Rule Size=5 considering a tradeoff between increasing the average BLEU score and not increasing the grammar size to a large extent. We report the results of varying these parameters in Table 3.3. We use the same parameters for all language pairs which have English as a target language. Other non-default decoder parameters that we used are: *max-chart-span=25*, the maximum span of a derivation and *cube-pruning-pop-limit=1000*, the number of hypotheses created for each chart span.

[1] The tokenization rules were adapted from `http://www.cis.upenn.edu/~treebank/tokenizer.sed` and integrated in the Moses tokenization script under the option -penn.

Depth	Nodes	Size	Grammar size	BLEU
3	15	3	2,572,222	19.17
4	20	4	3,188,970	19.30
5	**20**	**5**	3,668,205	**19.42**
5	30	5	3,776,961	19.42
5	30	6	4,340,716	19.43

Table 3.3: Evaluating the impact of the rule extraction parameters on grammar size and translation quality. Cased BLEU scores are averaged over newstest2009, 2010, 2011. This table was adapted from Nădejde et al. [2013].

3.3 Improving robustness

In this section we describe several methods of improving grammar coverage, making the string-to-tree system more robust. Firstly, we reduce the number of translation rules that drop the verb by re-aligning verbs. Secondly, we increase the number of extracted rules through restructuring constituency trees, thus reducing the structures that break heuristics regarding alignment. Thirdly, we improve the translation of named entities, which are often the main arguments of verbs, by avoiding that the compound splitting algorithm is applied to them.

3.3.1 Verb dropping

A major problem for German→English machine translation is the tendency to drop verbs. This is caused on the one hand by the failure of IBM models to properly align the German verb to its English equivalent because of the difference in word order. On the other hand, the verb might not be reordered in the translated sentence which can result in lower language model scores and BLEU scores. To reduce the number of translation rules which have a verb on the source-side but no aligned verb on the target-side, we propose the following method for realigning verbs prior to rule extraction. First the source and target verbs are identified by their part-of-speech tags. Then the un-aligned source verbs are aligned to the target verb for which IBM model 1 predicts the highest translation probability. Finally, using the updated alignment we extract new SCFG rules. Table 3.4 shows that when realigning verbs, the number of rules in the grammar that drop the target verb is almost three times lower and the output has more translated verbs. However, there is no change in BLEU scores.

		newstest2012		newstest2013	
System	Vb drop rules	#Vb	BLEU	#Vb	BLEU
Baseline	1,038,597	9,216	23.21	8,418	26.27
Realigned verbs	391,231	9,471	23.26	8,614	26.26
Reference translation	-	9,992	-	9,207	-

Table 3.4: Verb (Vb) dropping statistics. For each system we report the number of rules in the grammar that drop the verb on the target, the total number of verbs in the output (#Vb) and the BLEU score. This table was adapted from Nădejde et al. [2013].

3.3.2 Tree restructuring

Improving grammar coverage by restructuring constituency trees was previously explored for other language pairs [Wang et al., 2007]. Our contributions are to show that tree restructuring strategies are useful for the German→English language pair and to adapt the extraction parameters accordingly.

In this work we restructure the target constituency trees before rule extraction using *binarization*, which we describe in Section 2.2.2. Since binarization strategies increase the tree depth and number of nodes by adding virtual non-terminals, we increase the extraction parameters to: *Rule Depth = 7, Node Count = 100, Rule Size = 7*.

Table 3.5 shows the BLEU scores for the baseline system [Nădejde et al., 2013] and two systems employing different binarization strategies, as well as the resulting grammar sizes. The grammar size is computed after filtering out the rules that do not contain any of the source words in the evaluation sets, and would therefore not be used for translating these evaluation sets. Results show that the *right binarization* strategy improves translation quality more than *left binarization*, and also allows for more rules to be extracted.

3.3.3 Compound Splitting of Named Entities

Named entities, such as person and place names, often appear as arguments of the verb – for instance, as the subject or a prepositional modifier. While most named entities are not well represented in the training data, some have sub-units that are frequent words

	BLEU		Grammar size	
system	dev	test	dev	test
baseline	23.2	26.3	11,649,415	11.404.047
+ left binarization	23.5	26.4	21,779,125	21,387,976
+ right binarization	23.7	26.8	25,133,512	24,706,277

Table 3.5: German to English results on the dev (newsdev2012) and test (newstest2012) sets. Grammar sizes reported after filtering the rule table for the dev/test sets. This table was adapted from Williams et al. [2014].

which the compound splitter over-splits. For example, the script provided with Moses for compound splitting will split the city names in the following phrase *"Florstadt nach Bad Salzhausen"* into *"flor Stadt nach Bad Salz hausen"*. This is then wrongly translated by the baseline system as *"Flor after bath salt station"* instead of *"from Florstadt to Bad Salzhausen"*. To avoid over-splitting named entities, we apply a 3–class named entity tagger [Finkel et al., 2005, Faruqui and Padó, 2010] on the German side of the corpus prior to splitting. By marking the named entities (persons, locations, organizations) we prevent the compound splitter from splitting these. We remove the annotations after compound splitting and prior to rule extraction.

	BLEU	
system	newstest2015	newstest2016
baseline	28.6	33.5
+ NER before split	28.8	33.8

Table 3.6: German→English translation results when avoiding over-splitting of named entities by running named entity recognition (NER) before splitting. This table was adapted from Williams et al. [2016].

In Table 3.6 we compare a baseline German→English string-to-tree system with the system that does not over-split named entities. Both system are trained with right binarization of the parse trees and with the following non-default parameter value: *max-chart-span = 50*. This limits sub derivations to a maximum span of 50 source words. In addition we use sparse features to determine the non-terminal labels for unknown words similar to the English→German systems described by Williams et al. [2014] and Sennrich et al. [2015]. The results show that annotating named entities before splitting leads to an improvement of 0.3 BLEU . For future work, we propose

	BLEU	
system	newstest2014	newstest2015
Hiero	27.7	28.0
String-to-tree	28.7	28.7
+ bilingual NLMs	28.6	28.7

Table 3.7: German→English translation results with bilingual neural language models (NLMs). This table was adapted from Williams et al. [2015].

to manually evaluate the translation accuracy of named entities and investigate to what extent is the accuracy of the named entity recognizer impacting translation peformance.

3.3.4 Neural Bilingual Language Models

In the previous sections we presented several methods for improving grammar coverage. Still, the translation model does not generalize well since it is estimated by counting and the translation rules are limited to just a few lexical items. Neural bilingual language models (BiNNLMs) improve generalization by representing the words in a continuous vector space. These models are also able to condition target words on a larger source context, even if that particular source context does not appear in the training data. In this section we present experiments with BiNNLMs, which were previously reported in Williams et al. [2015].

The bilingual language models are trained with the NPLM toolkit [Vaswani et al., 2013]. We use 250-dimensional input embedding and the hidden layer, and input and output vocabulary sizes of 500,000 and 250,000 respectively. The first BiNNLM is a 5-gram model with an additional context of 9 source words, the affiliated source word and a window of 4 words on either side. The second second model is a 1-gram model with an additional context of 13 source words.

In Table 3.7 we compare the Hiero system[2] [Chiang, 2007], the baseline string-to-tree system and the string-to-tree system augmented with the two bilingual neural language models. The Hiero system overgeneralizes in terms of re-ordering and performs worse than the string-to-tree system which produces more grammatical translations. Adding the BiNNLMs did not improve the BLEU scores. Birch et al. [2014] previously reported that the BiNNLMs, trained with the same hyper-parameters, did not improve the phrase-based system for the German→English translation task at IWSLT2014.

[2] For the Hiero baseline system we use *max-chart-span = 15*.

Similarly, Haddow et al. [2015] report that the BiNNLMs did not improve the phrase-based system for the French→English news translation task at WMT2015. Our result for the string-to-tree system support previous observations, that BiNNLMs are not helpful when translating into English, a morphologically poor language, for which strong n-gram LMs are available.

3.4 Error Analysis

In this section, we analyze the output of a string-to-tree system and identify specific errors that we address in the rest of this thesis. Firstly, we present the results of the evaluation of a string-to-tree system using the HMEANT [Lo and Wu, 2011] metric, showing that the semantic frames are not translated accurately. Secondly, we perform an error analysis to determine which co-occurring errors in the translation and its syntactic parse tree lead to the mis-translation of the semantic frames. This analysis shows that the string-to-tree system has problems with prepositional phrase (PP) and noun phrase (NP) attachment, and also with verb translation.

3.4.1 HMEANT evaluation

Machine translation output is most commonly evaluated with automatic metrics such as BLEU [Papineni et al., 2002], METEOR [Lavie and Denkowski, 2009] and TER [Snover et al., 2009]. These metrics compute word overlap or compare shallow surface properties between machine translated output and reference translations. However, they are not able to capture how much of the meaning of the source sentence is retained in the machine translation output. HMEANT [Lo and Wu, 2011] is a human evaluation metric which proposes to measure the semantic overlap between the translation and the reference by looking at the semantic frames of predicates.

Annotators are tasked with identifying and labeling the verbal predicates in a sentence and their corresponding semantic roles, in both the translation and the reference. Next, the annotator tries to align the predicates and then the corresponding semantic roles. The HMEANT metric computes an f-score from the counts of correctly aligned predicates and semantic roles between the translations and references. The HMEANT score computed over a test set should indicate how good the system is at translating the core information in the source sentence such as *who did what* to *who*, *when*, *where* and *why*.

In Birch et al. [2013], we investigated how reliable HMEANT is for evaluating MT. The German→English string-to-tree system we included in the analysis was described in Section 3.2. The overall HMEANT scores averaged across annotators and sentences, presented in Table 3.8, show that the string-to-tree system is better at translating the semantic frame of verbs than the rule-based or phrase-based systems. The improved HMEANT score can be attributed to the system translating more predicates. Nonetheless, all systems perform poorly at retaining the meaning of the source sentence.

Language Pair	System	BLEU	HMEANT	Aligned predicates
German→English	Phrase–based	35.1	63.4	234
	String–to–tree	34.4	66.7	245
	Rule–based	29.5	62.5	236

Table 3.8: Comparison of HMEANT score averaged across sentences and annotators and (smoothed sentence) BLEU score for three MT systems. This table was adapted from Birch et al. [2013].

We make a few observations regarding the process of identifying and matching the semantic frames, which we considered in the rest of this thesis. Firstly, only verbs are marked as the head of a semantic frame. Therefore many semantic frames were not annotated because the verb was missing or because the head of the semantic frame was a noun. Secondly, prepositional phrases attached to a noun can often change the semantics of a sentence.

Other issues of interest for modeling semantic information in string-to-tree systems are the embedding of semantic frames within other frames and that identifying and matching the semantic roles between reference and machine translation output is hard even for humans.

In this thesis, we opted to use dependency relations to represent the predicate-argument structure, motivated by these challenges and the modest success of integrating semantic role labels (SRL) with SMT [Wu and Fung, 2009a, Li et al., 2013]. Dependency parsers are more accurate and robust than SRL parsers and are also available for a multitude of languages [Nivre et al., 2016]. In Chapter 4 we use dependency relations to model semantic affinities between target predicates (verbal and nominal) and their core and prepositional modifiers. In Chapter 5 we use dependency relations to identify the subcategorization frame and arguments of the source verbs, which are then used as context to predict the correct lexical choice for the target verb.

Category	Description
PP Attachment	Incorrectly attached prepositional phrase
NP Attachment	Incorrectly attached noun phrase
Modifier Attachment	Incorrectly attached adjectives and adverbs
Clause Attachment	Transformation involving an S node
NP Structure	Transformations involving Noun Phrase structure
VB	Errors involving missing or mistranslated verbs
Reordering	Errors involving word reordering
Surface form	Errors in the surface form, when meaning is lost

Table 3.9: Error types considered in manual error analysis of string-to-tree MT system.

3.4.2 Manual Error Analysis

Before ending the chapter, in this section we present an error analysis of the target syntactic structure generated by the string-to-tree system for German→English. Identifying the most frequent errors allows us to deconstruct the translation of predicate-argument structures into smaller problems, and informs the design of the models presented in the following two chapters.

Ideally, error detection should be done automatically similarly to the analysis of monolingual syntactic parsers performed by the Berkeley Parser Analyser[3] [Kummerfeld et al., 2012a]. This analyser provides linguistic intuitions about different types of parsing errors. The output of a syntactic parser is transformed into the human-annotated gold standard parse tree and the transformations are classified into error types. However, automatic classification of error types in the output of string-to-tree systems is hard, since there are many acceptable translations with possibly distinct syntactic structures. Providing gold annotated parse trees for each acceptable translation would be time consuming. Instead, we conduct a manual error analysis of the translations and the corresponding syntactic trees produced by the string-to-tree system. We describe the error categories in Table 3.9, grouping the categories proposed by Kummerfeld et al. [2012a] above the line and the categories that correlate the syntactic errors with the mistranslation of the predicate-argument structures below the line.

We conducted the error analysis on 50 sentences randomly selected from the WMT 2013 test set [4]. The total number of errors for each category is reported in Table 3.10.

[3] Berkeley Parser Analyserhttp://code.google.com/p/berkeley-parser-analyser/

[4] A subset of the sentences used for the HMEANT evaluation.

Error type	Error count	% of sentences
PP attachment	20	30
NP attachment	28	38
Modifier attachment	22	34
Clause attachment	17	24
NP structure	12	22
VB	26	42
Reordering	29	42
Surface form	31	44

Table 3.10: Error count and percentage of sentences affected by each type of error category. Results aggregated over 50 sentences.

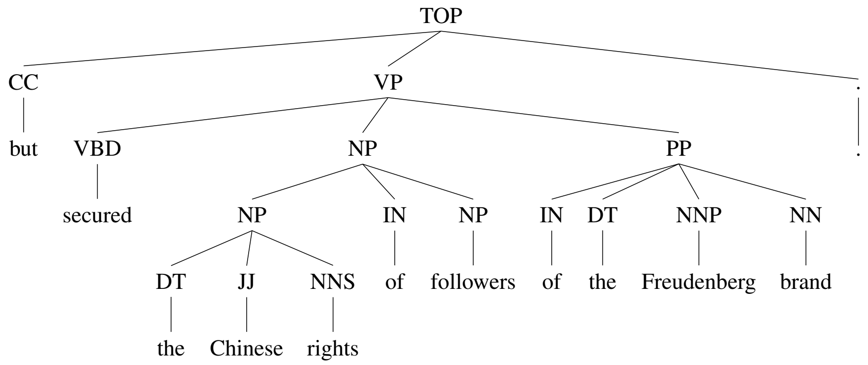
In the case when a sentence has multiple errors, for example an attachment error and a reordering error, we count all the errors. Therefore, the percentages reported in the third column do not sum to 100%.

Although we counted many syntactic errors involving attachment, not all generate surface form errors. Surface form errors are caused by a combination of verb errors or attachment errors and reordering errors. However, even though the surface form may be correct, the attachment errors can negatively impact the performance of target-side features such as the selectional preference model (Chapter 4) and the neural verb lexicon model (Chapter 5). The neural verb lexicon model addresses the problem of mistranslated or missing verbs, which affects almost half of the sentences. To improve translation of NP and PP modifiers, we propose the selectional preference model capturing the semantic affinities between predicates and their arguments.

Next we give examples of mistranslated predicate-argument structures, and comment on how the errors considered in the manual analysis relate to incomplete subcategorization frames and to mistranslated or wrongly attached arguments.

In the example in Figure 3.1 the semantic frame of the verb *"secured"* is mistranslated. The subcategorization frame is incomplete because the subject is mistranslated and attached to a noun phrase instead of the verb. The semantic frame should have the following interpretation: "*[The plagiarizer]$_{AGENT}$* secured [the Chinese rights to the Freudenberg brand]$_{PATIENT}$". For this example we counted the following errors:

- PP attachment – PP "*of the Freudenberg brand*" is wrongly attached. It should



Source:	*"Doch sicherte sich der Nachahmer weiterhin die chinesischen Rechte an der Marke Freudenberg."*
Reference:	*"However, the plagiarizer still secured the Chinese rights to the Freudenberg brand."*

Figure 3.1: Example of mistranslated semantic frame for the verb "sicherte (secured)".

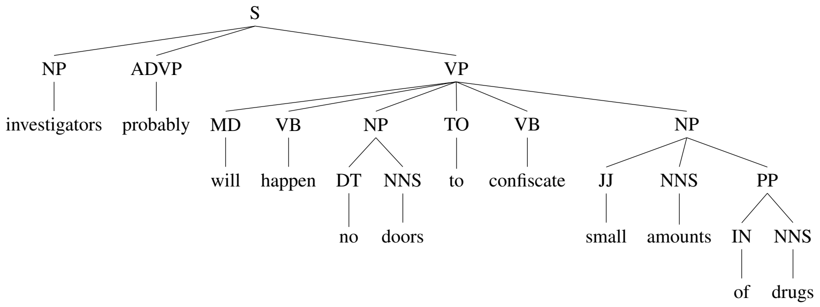
modify the NP "*the Chinese rights*".

- NP attachment – NP "*followers*" is mistranslated and wrongly attached to the NP "*the Chinese rights*". It should be translated as "*plagiarizer*" and attached as the subject of the verb "*secured*".
- Reordering – NP "*followers*" should come before the verb "*secured*".
- Surface form – The semantic frame of the verb "*secured*" is mistranslated.

In the example shown in Figure 3.2, the meaning is lost because the source verb *"eintreten"* is mistranslated as *"happen"* instead of *"beat down"*. The language model context was not enough to disambiguate the verb. In addition, an NP attachment error occurs because *"small bag of drugs"* should be attached to the verb *"confiscate"* to form another verb phrase.

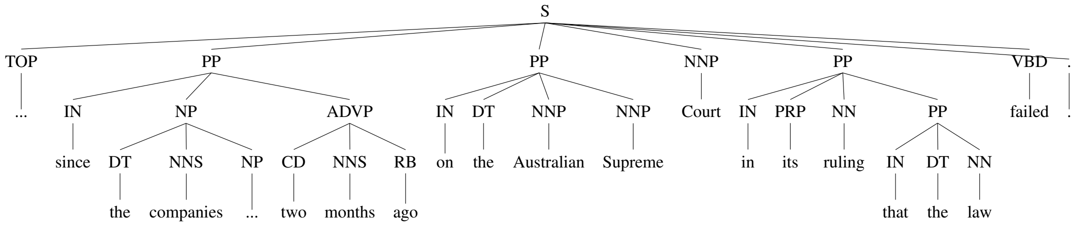
A final example is shown in Figure 3.3, where the entire sub-categorization frame of the verb is mistranslated. The verb *"scheiterten"* is mistranslated as *"failed"* instead of *"lost"* and its arguments are either in a wrong order or mistranslated.

The semantic frame should have the following interpretation:
"[*The Companies*]$_{AGENT}$ *lost* [*a challenge to the laws*]$_{PATIENT}$ [*two months ago*]$_{TEMP}$ [*in the Australian Supreme Court*]$_{LOC}$". The system generated many prepositional phrases which are underspecified with respect to their syntactic function and semantic role. The system does not have access to semantic information to distinguish the subject and the object, or to assign the correct order of the arguments. For this example



Source	*“Kriminalbeamte zwar wahrscheinlich keine Türen eintreten werden , um kleinere Mengen der Drogen zu beschlagnahmen.”*
Reference	*“Drug agents will probably not beat down doors to seize a small bag of the drug.”*

Figure 3.2: Example of a mistranslated verb: *“eintreten (beat down)”*.



Source	*“... seit die Konzerne ... vor zwei Monaten am obersten australischen Gerichtshof mit ihrer Anfechtung der Gesetze scheiterten.”*
Reference	*“ ... two months ago since the companies ... lost a challenge to the laws in Australia’s high court.”*

Figure 3.3: Example of mistranslated semantic frame for the verb *“scheiterten (lost)”*.

we counted the following errors: one modifier attachment, one PP attachment and two NP attachment.

3.5 Conclusion

In this chapter we presented several methods for improving the robustness of string-to-tree systems. In the baseline system, we addressed issues regarding tokenization and rule extraction parameters. Then we improved grammar coverage using tree restructuring, verb re-alignement and pre-processing of named entities. We also explored improving generalization with neural language models. Finally, the error analyses showed that the string-to-tree systems have problems with translating the semantic frames of verbs and more specifically with the attachment of noun phrases and prepositional phrases. Based on the issues identified with these analyses, we propose using dependency relations to model semantic fitness between predicates and their arguments (Chapter 4), as well as to identify relevant source context for improving translation of verbs (Chapter 5).

Chapter 4

A Selectional Preferences Model

4.1 Introduction

In the previous chapter, we showed that string-to-tree systems can achieve state-of-the-art results for German→English, a language pair exhibiting long distance word reordering. Still, according to the error analyses, even a competitive string-to-tree system does not translate the semantic frames accurately. In this chapter, we explore whether knowledge about semantic affinities between the target predicates and their argument fillers is useful for translating ambiguous predicates and arguments. We use the term *semantic affinity* to refer to the co-occurrence behavior of the predicates and the semantic classes of their arguments, which we quantify using the selectional association measure of Resnik [1996].

We propose a selectional preferences feature based on the selectional association measure of Resnik [1996] and integrate it in a string-to-tree decoder. The feature models selectional preferences of verbs for their core and prepositional arguments as well as selectional preferences of nouns for their prepositional arguments.

In previous work [Wu and Fung, 2009a, Liu and Gildea, 2010, Li et al., 2013] Semantic Role Labels (SRL) were used to represent the predicate-argument structures. However, this representation poses the following problems: SRL do not cover all the words in a sentence, SRL parsers have lower accuracy than syntactic parsers, and it is hard to integrate SRL directly in the decoder. Furthermore, SRL for nominal predicates is less accurate than for verbal predicates. For these reasons, we prefer dependency relations as a syntactic-semantic interface to representing the verbal and nominal predicates and their arguments.

We present a contrastive evaluation of syntactic representations showing that a string-to-tree system with target-side dependency relations is competitive when compared to the string-to-tree system with target-side phrase-structures, described in the previous chapter. We then use a dependency-based string-to-tree system as baseline and build the selectional preferences feature on top of the target-side dependencies.

We compare our feature with a variant of the neural relational dependency language model (RDLM) [Sennrich, 2015] and find that neither of the features improves automatic evaluation metrics. For one variant of the proposed feature, we found a slight improvement in automatic evaluation metrics when translating short sentences as well as an increase in precision for verb translation. We conclude that mistranslated verbs and errors in the target syntactic representation produced by the decoder are negatively impacting these features.

This chapter is structured as follows. In Section 4.2 we describe why string-to-tree systems may translate the predicates or their arguments incorrectly and survey prior work that addresses this. In Section 4.3 we present a comparison of target-side representations for string-to-tree systems. In Section 4.4 we formally describe the selectional preference feature for dependency-based string-to-tree systems. Section 4.5 describes the experimental setup and Section 4.6 presents the results of the automatic evaluation, as well as a qualitative analysis of the machine translated output.

4.2 Background

String-to-tree systems suffer from errors such as scrambled or mis-translated predicate-argument structures, which are reflected in the HMEANT evaluation presented in Section 3.4.1 . We give examples of mistranslated verbal and nominal predicates in Table 4.1.

In example a) the baseline system MT1 mistranslates the verb *"besichtigt"* as *"viewed"*. The system MT2 which uses information about the semantic affinity between the verb and its argument produces a better translation: *"visited"*. The semantic affinity is quantified using the *selectional association* measure [Resnik, 1996]. The score shown on the right, for the verb *"visited"* and argument *"trip"* in the syntactic relation *prep_on* [1] is indicating a stronger affinity than for the baseline translation. In example b) the baseline system MT1 mistranslates the noun *"Aufnahmen"* as *"recordings"* while the system MT2 produces the correct translation *"images"* which is a better fit for the prepositional modifier *"from the telescope"*. One error that both sys-

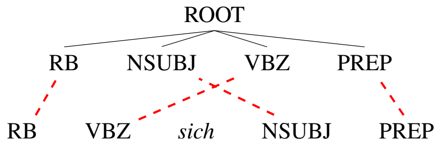
[1] Prepositional modifier with the syntactic head *"on"*.

			(relation, predicate, argument)	SelAssoc
a)	SRC	Bei nur einer Reise können nicht alle davon *besichtigt* werden.		
	REF	You won’t be able to *visit* all of them on one trip .		
	MT1	Not all of them can be *viewed* on only one trip.	(prep_on, *viewed*, trip)	-0.154
	MT2	Not all of them can be *visited* on only one trip.	(prep_on, *visited*, trip)	1.042
b)	SRC	Eine der schärfsten *Aufnahmen* des Hubble-Teleskops		
	REF	One of the sharpest *pictures* from the Hubble telescope		
	MT1	One of the strongest *recordings* of the Hubble telescope	(prep_of, *recordings*, telescope)	-0.0004
	MT2	One of the strongest *images* from the Hubble telescope	(prep_from, *images*, telescope)	0.3917

Table 4.1: Examples of errors in the predicate-argument structure produced by a string-to-tree MT system. a) mistranslated verbal predicate b) mistranslated nominal predicate. Selectional association (SelAssoc) scores are shown on the right. Higher scores indicate a stronger semantic affinity. Negative scores indicate a lack of semantic affinity.

tems make is to translate *“schärfsten”* as *“strongest”* instead of *“sharpest”*. While this is also a semantic error, we do not address the problem of selectional preferences of adjectives with our feature.

String-to-tree MT systems handle long distance reordering with synchronous CFG rules such as the rule in Figure 4.1.



$$\text{ROOT} \rightarrow \text{RB}_0\ \text{NSUBJ}_1\ \text{VBZ}_2\ \text{PREP}_3\ |||\ \text{X}_0\ \text{X}_2\ \textit{sich}\ \text{X}_1\ \text{X}_3$$

Figure 4.1: Reordering translation rule. The target syntactic sub-tree and the alignment of the non-terminals to the source-side spans is depicted at the top. The corresponding synchronous context free grammar rule is depicted at the bottom.

The figure shows a target sub-tree with the alignment between the target non-terminals and the corresponding source spans. The non-terminals are either part-of-speech labels or dependency relations and the mapping from source to target non-terminals is indicated in the synchronous CFG rule by the subscript numbers. This synchronous CFG rule reorders the verb and its arguments according to the target side word order. However, it does not contain the lexical heads of the verbal predicate, the subject or the prepositional modifier. Therefore, the entire predicate argument structure is translated by subsequent independent rules. The language model context will capture at most the verb and one main argument. Due to the lack of a larger source or target context the resulting predicate-argument structures are often not semantically coherent.

From a syntactic perspective, a correct predicate-argument structure will have the sub-categorization frame of the predicate filled in. Weller et al. [2013] use sub–categorization information to improve case-prediction for noun phrases when translating into German. Case prediction for noun phrases is important in the German language as it indicates the grammatical function. Their approach however did not produce strong improvements over the baseline. From a large corpus annotated with dependency relations, they extract verb-noun tuples and their associated syntactic functions: direct object, indirect object, subject. They also extract triples of verb-preposition-noun in order to predict the case of noun-phrases within prepositional-phrases. The probabilities of such tuples and triples are computed using relative frequencies and then used as a feature for a CRF classifier that predicts the case of noun-phrases. Weller et al. [2013] apply the CRF classifier to the output of a word-to-stem phrased-based translation system as a post-processing step. In contrast, our model is used directly as a feature in the decoder. While Weller et al. [2013] identify the arguments of the verb and their grammatical function by projecting the information from the source sentence we use the dependency relations produced by the string-to-tree decoder. We also consider prepositional modifiers of nouns.

Weller et al. [2014] propose using noun class information to model selectional preferences of prepositions in a string-to-tree translation system. They use the noun class information to annotate PP translation rules in order to restrict their applicability to specific semantic classes. In our work we don't impose hard constraints on the translation rules, but rather soft constraints using our model as a feature in the decoder. While we use word embeddings to cluster arguments, Weller et al. [2014] experiment with a lexical semantic taxonomy and clustering words based on co-occurrences within a window or syntactic features extracted from dependency-parsed data.

Modeling reordering and deletion of semantic roles [Wu and Fung, 2009a, Liu and Gildea, 2010, Li et al., 2013] has been another line of research on improving translation of predicate-argument structures. For tree-to-string SMT, Liu and Gildea [2010] propose modeling reordering of the source-side semantic frame and for hierarchical SMT, Li et al. [2013] propose finer grained features that distinguish between predicate-argument reordering and argument-argument reordering. Gao and Vogel [2011] and

Bazrafshan and Gildea [2013] annotate target non-terminals with the semantic roles they cover in order to extract synchronous grammar rules, for hierarchical SMT and string-to-tree SMT respectively, that cover the entire predicate argument structure. In contrast, our work models lexical semantic affinities, and not reordering, between target predicates and their arguments in string-to-tree SMT.

Following our work, Tang et al. [2016] compare different selectional preference models for hierarchical[2] SMT. They propose modeling selectional preferences of verbs for their direct objects and subjects using conditional probabilities over the head words. In our work we also consider nominal predicates and prepositional modifiers. Different from their work, we use a baseline translation system which encodes local target-side syntactic dependencies in the translation rules. The models proposed by Tang et al. [2016] are a subclass of the Relational Dependency Language Model [Sennrich, 2015] which we use in our contrastive experiments.

We propose using selectional preference over predicate and arguments in the target as this is a simple way of leveraging external knowledge in the string-to-tree framework.

4.3 Target–side Dependency Relations

In this section we compare two target-side syntactic representations for the string-to-tree system: PTB–style phrase-structures and dependency relations. We determine to what extent the dependency representation impacts the quality of a baseline string-to-tree system, as we will use dependencies to represent the predicate-argument structure and model target-side semantic affinities. As dependency representation we use the collapsed form of Stanford typed dependencies [de Marneffe and Manning, 2008], which we described in Section 2.5. We present a contrastive evaluation between a string-to-tree system with PTB–style phrase-structures, described in detail in Chapter 3, and a string-to-tree system with Stanford typed-dependencies as target syntax. These experiments were also previously reported in Williams et al. [2016].

We train the baseline string-to-tree systems with phrase-structures as target syntax, to which we apply right binarization, for German→English and Romanian→English. For the dependency-based string-to-tree systems, the English side of the parallel corpora is annotated with the Stanford Neural Network dependency parser [Chen and Manning, 2014b]. The resulting dependency trees are then converted to constituency

[2] Phrase-based bracketing transduction grammar (BTG).

trees as described in Section 2.5, followed by head-binarization [Sennrich and Haddow, 2015]. This conversion allows us to extract a new grammar for the dependency-based string-to-tree system using the GHKM algorithm and the target-side dependency relations. For Romanian→English we allow glue rules, described in Section 2.2.1.1, and normalize the corpora by removing all diacritics from the Romanian side. The size of the training, tuning and test sets are reported in Table 4.2.

Language pair	Train	Tune	Test
DE-EN	4,494,661	2,000	2,999
RO-EN	608, 315	999	1,999

Table 4.2: Corpus statistics for parallel data. The number of sentences is reported for the WMT16 datasets.

We report the cased BLEU scores for the different setups of the string-to-tree system in Table 4.3. We also measure the effect of extraction parameters, which control the size of the extracted SCFG rules and of the resulting grammar, on string-to-tree systems with target dependencies. The dependency-based system performs worse than the baseline for lower values of the *Nodes* parameter. However, when setting this parameter as for the baseline the performance is comparable. We conclude that a competitive string-to-tree system can be trained using dependencies as target-side syntax and head-binarization.

	Parameters	Binarization	RO→EN		DE→EN	
Syntax	(Nodes, Depth, Size)		dev	test	dev	test
Phrase-structure	(100, 7, 7)	Right	34.2	33.0	28.8	33.8
Dependency	(40, 7, 7)	Head	33.7	32.3	28.1	33.0
Dependency	(100, 7, 7)	Head	34.3	33.2	-	-

Table 4.3: Comparison of string-to-tree systems with phrase-structures and dependency relations as target-syntax. Cased BLEU score reported on dev (half of newsdev2016) and test (newstest2016).

4.4 Selectional Preference Feature

In this section we introduce the concept of *Selectional Preferences* and a data-driven information theoretic measure for these. We then describe how we adapt this measure to implement it as a feature in the string-to-tree decoder and how our proposed feature compares to the relational dependency language model [Sennrich, 2015].

4.4.1 Learning Selectional Preferences

Selectional preferences describe the semantic affinities between predicates and their argument fillers. For example, the verb *"drinks"* has a strong preference for arguments in the conceptual class of *"liquids"*. Therefore the word *"wine"* can be disambiguated when it appears in relation to the verb *"drinks"*. A corpus driven approach to modeling selectional preferences usually involves extracting triples of (*syntactic relation, predicate, argument*) and computing co-occurrence statistics. The predicate and argument are represented by their head words and the triples are extracted from automatically parsed data. Another typical step is generalizing over seen arguments. Approaches to generalization include using an ontology such as WordNet [Resnik, 1996], using distributional semantics similarity [Erk et al., 2010, Séaghdha, 2010, Ritter et al., 2010], clustering [Sun and Korhonen, 2009], multi-modal datasets [Shutova et al., 2015], and neural networks [Cruys, 2014].

We base our feature on the *selectional association* measure proposed by Resnik [1996], which in turn is defined based on the *selectional preference* measure. The information theoretic selectional preference measure quantifies the difference between the posterior distribution of an argument class given the verb and the prior distribution of the class. For instance, *"person"* has a higher prior probability than *"insect"* to appear in the subject relation, but, knowing the verb is *"fly"*, the posterior probability becomes higher for *"insect"*. In our work we use clusters, instead of word classes, to generalized over unseen arguments.

Resnik's formal definition of the *selectional preference* of a predicate is:

$$\begin{aligned} SelPref(p,r) &= KL(P(c|p,r) \parallel P(c|r)) \\ &= \sum_{c} P(c|p,r) log \frac{P(c|p,r)}{P(c|r)} \end{aligned} \tag{4.1}$$

where KL is the Kullback - Leibler divergence, r is the relation type, p is the predicate and c is the conceptual class of the argument. Resnik uses WordNet to obtain the conceptual classes of arguments, therefore generalizing over seen arguments.

The *selectional association* or semantic affinity between a predicate and an argument class is quantified as the relative contribution of the class towards the overall selectional preference of the predicate:

Verb	Relation	SelPref	Argument	SelAssoc
see	dobj	0.56	PRN	0.123
			movie	0.022
			episode	0.001
is–hereditary	nsubj	1.69	disease	0.267
			monarchy	0.148
			title	0.082
drink	dobj	3.90	water	0.144
			wine	0.061
			glass	0.027

Table 4.4: Example of *selectional preference* (SelPref) and *selectional association* (SelAssoc) scores for different verbs. PRN is the class of pronouns. This table was adapted from Nădejde et al. [2016a].

$$SelAssoc(p,r,c) = \frac{P(c|p,r)log\frac{P(c|p,r)}{P(c|r)}}{SelPref(p,r)} \tag{4.2}$$

We give examples of the *selectional preference* and *selectional association* scores for different verbs and their arguments in Table 4.4. The verb *"see"* takes on many arguments as direct objects and therefore has a lower selectional preference strength for this syntactic relation. In contrast, the predicate *"hereditary"* takes on fewer arguments for which it has a stronger selectional preference.

Several selectional preference models have been used as features in discriminative syntactic parsing systems. Cohen et al. [2012] observe that when parsing out-of-domain data many attachment errors occur for the following syntactic configurations: head (V or N) – prep – obj and head (N) – adj. The authors proposed a class-based measure of selectional preferences for these syntactic configurations and learn the argument classes using Latent Dirichlet Allocation (LDA). Kiperwasser and Goldberg [2015] compare different measures of lexical association between head word and modifier word for improving dependency parsing. Their results show that the association measure based on pointwise mutual information (PMI) has similar generalization capabilities as a measure of distributional similarity between word embeddings. van Noord [2007] has shown that bilexical association scores computed using PMI for all types of dependency relations are a useful feature for improving dependency parsing in Dutch.

Following our work, Tang et al. [2016] show improvements by modeling selectional preferences of verbs for their direct objects and subjects, in hierarchical SMT, using conditional probabilities over the words. Their models capture a subset of the relations we consider and are a subclass of the Relational Dependency Language Model which we denote with RDLM–P_w in section 4.6.3. The authors also propose a cross-lingual variant of their model which conditions the argument on the corresponding source verb.

4.4.2 Adaptation of Selectional Preference Models for Syntax-Based Machine Translation.

We are interested in modeling selectional preferences of verbs for their core and prepositional arguments as well as selectional preferences of nouns for their prepositional arguments. We identify the relation between a predicate and its argument from the dependency tree produced by a string-to-tree machine translation system. Since we are interested in using the feature during decoding, we need the model to be fast to query and have broad coverage.

Our selectional preference feature is a variant of the information theoretic measure of Resnik [1996] defined in Eq 4.2. While Resnik uses the WordNet classes of the arguments, this is not appropriate for a machine translation task where the vocabulary has millions of words and English is not the only targeted language. Therefore, we adapt Resnik's selectional association measure in two ways:

- In the first model *SelAssoc_L* we compute the co-occurrence statistics defined in Eq 4.1 and Eq 4.2 over lemmas of the predicate and argument head words.
- In the second model *SelAssoc_C* we replace the WordNet classes in Eq 4.1 and Eq 4.2 with word clusters[3]. We obtain the 500 word clusters by applying the k-means algorithm to the glovec word embeddings [Pennington et al., 2014].

Prepositional phrase attachment remains a frequent and challenging error for syntactic parsers [Kummerfeld et al., 2012b] and translation of prepositions is a challenge for SMT [Weller et al., 2014]. Therefore we decide to train two separate models, each with its own weight in the log-linear model: one for main arguments (*nsubj, nsubjpass, dobj, iobj*) and one for prepositional arguments.

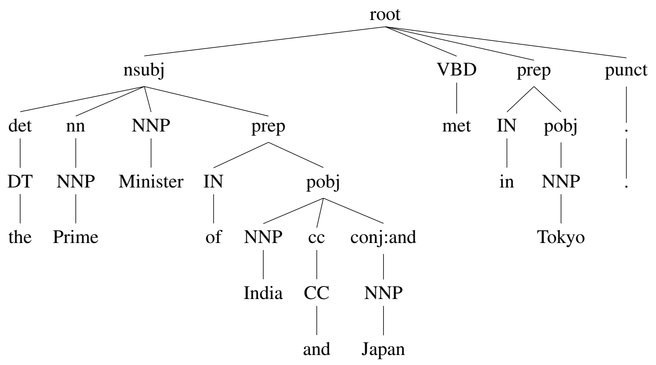
[3] We have not done experiments with WordNet classes.

4.4.3 Comparison with a Neural Relational Dependency Language Model.

Sennrich [2015] introduced a relational dependency language model (RDLM) for string-to-tree machine translation, trained with a feedforward neural network. For a sentence S with symbols $w_1, w_2, ...w_n$ and dependency labels $l_1, l_2, ...l_n$ with l_i the label of the incoming arc at position i, RDLM is defined as:

$$\begin{aligned} P(S,D) &\approx \prod_{i=1}^{n} P_l(i) \times P_w(i) \\ P_l(i) &= P(l_i \mid h_s(i)_1^q, l_s(i)_1^q, h_a(i)_1^r, l_a(i)_1^r) \\ P_w(i) &= P(w_i \mid h_s(i)_1^q, l_s(i)_1^q, h_a(i)_1^r, l_a(i)_1^r, l_i) \end{aligned} \tag{4.3}$$

where for each of q siblings and r ancestors of w_i, h_s and h_a are their head words and l_s and l_a their dependency labels. The $P_w(i)$ distribution models similar information as our proposed feature *SelAssoc*. However, we use only $h_a(i)_1, l_i$ as context and consider only a subset of dependency labels: *nsubj, nsubjpass, dobj, iobj, prep*. The reduced context alleviates problems of data sparsity and is more reliably extracted at decoding time. The subset of dependency relations identify arguments for which predicates might exhibit selectional preferences. Our feature is different from $RDLM - P_w$ as it quantifies the difference between the posterior distribution of an argument class given the verb and the prior distribution of the argument class. We hypothesize that such information is useful when translating arguments that appear less frequently in the training data but are also prototypical for certain predicates. For example the triples *(bus, drive, dobj)* and *(van, drive, dobj)* have the following log posterior probabilities and *SelAssoc* scores: log P(bus | drive, dobj) = -5.44, log P(van | drive, dobj)= -5.58 and SelAssoc(bus, drive, dobj) = 0.0079, SelAssoc(van, drive, dobj) = 0.0103. The verb *drive* co-occurs less frequently with the direct object *van* than with *bus*, although both are prototypical arguments. However, *van* occurs more often with the verb *drive* than with other verbs and this is quantified by the selectional association measure. The selectional association score is higher for *van* than it is for *bus*, which occurs frequently with other verbs such as *ride, catch*.



relation	predicate	argument
nsubj	met	Minister
prep_in	met	Tokyo
prep_of	Minister	India

Figure 4.2: Example of a translation and its dependency tree in constituency representation produced by the string-to-tree statistical MT system. Triples extracted during decoding are shown on the right.

4.5 Experimental setup

We train a baseline dependency-based string-to-tree system for German→English on all available data provided at WMT15 [4] [Bojar et al., 2015]. The number of sentences in the training, tuning and test sets are shown in Table 4.5.

Train	Tune	Test
4,472,694	2,000	8,172

Table 4.5: Number of sentences for WMT15 dataset. The test set consists of the newstest2013, 2014 and 2015 corpora.

The English side of the parallel corpus is annotated with dependency relations using the Stanford dependency parser [Chen and Manning, 2014a]. To identify more accurately the *(dependency relation, predicate, argument)* triples at decoding time, we *do not* restructure target trees with head-binarization. The basic constituency representation allows to easily extract the head words of the predicate and argument, as each node has exactly one pre-terminal child which corresponds to the head word of that span. We use the following rule extraction parameters: *Rule Depth* = *5, Node Count* = *20, Rule Size* = *5*, we give a high penalty to glue rules and allow non-terminals to span a maximum of 50 words.

[4] We use a slightly different setup then in the Section 4.3, as the experiments in this section were conducted/published before.

For training the selectional preference features we extract triples of *(dependency relation, predicate, argument)* from parsed data, where the predicate and argument are identified by their head word. We use the English side of the parallel data and the Gigaword v.5 corpus parsed with Stanford typed dependencies [Napoles et al., 2012]. We use Stanford dependencies in the collapsed version, described in Section 2.5, which resolves coordination[5] and collapses the prepositions. Figure 4.2 shows an example of a translated sentence, its dependency tree produced by the string-to-tree system and the triples extracted at decoding time. We consider the following main arguments: *nsubj, nsubjpass, dobj, iobj* and *prep* arguments attached to both verbs and nouns. Table 4.6 shows the number of extracted triples.

Type of relation	Number of triples
main	540,109,283
prep	810,118,653
nsubj	315,852,775
nsubjpass	32,111,962
dobj	188,412,178
iobj	3,732,368

Table 4.6: Number of relation triples extracted from parsed data. The data consists of the English side of the parallel data and Gigaword. *main* arguments include: nsubj, nsubjpass, dobj, iobj.

We integrate the feature in a bottom-up chart decoder. The feature has several scores:

- A counter for the dependency triples covered by the current hypothesis.
- A selectional association score aggregated over all main arguments: nsubj, nsubjpass, dobj, iobj.
- A selectional association score aggregated over all prepositional arguments with no distinction between noun and verb modifiers.

For both tuning and evaluation of all machine translation systems we use a combination of the cased BLEU score and head-word chain metric (HWCM) [Liu and Gildea,

[5] Coordination is not resolved at decoding time.

2005]. The HWCM metric implemented in the Moses toolkit computes the harmonic mean of precision and recall over head-word chains of length 1 to 4. The head-word chains are extracted directly from the dependency tree produced by the string-to-tree decoder and from the parsed reference. Tuning is performed using batch MIRA [Cherry and Foster, 2012] on 1000-best lists. We report evaluation scores averaged over the newstest2013, newstest2014 and newstest2015 data sets provided by WMT15.

4.6 Evaluation

4.6.1 Error analysis

First, we analyse how often the verb and its arguments are mistranslated by the baseline dependency-based string-to-tree system. For this purpose we manually annotated errors in sentences with more than 5 words and at most 15 words. With this criterion we avoid translations with scrambled predicate-argument structures. Almost all sentences have one main verb.

To have a more reliable error annotation we first post-edited 100 translations from the baseline system. We then compared the translations with their post-editions and annotated error categories using the BLAST tool [Stymne, 2011]. We considered a *sense* error category when there was a wrong lexical choice for the head of a main argument, a prepositional modifier or the main verb. We also annotated mistranslated prepositions.

Error Category	Error Count	Total
Preposition	18	143
Sense	53	388
Main argument	18	145
Prep modifier	9	143
Main verb	26	100

Table 4.7: Number of mistranslated words in 100 sentences manually annotated with error categories. The "Sense" error category refers to mistranslated content words and is further broken down by part of sentence containing the issue. This table was adapted from Nădejde et al. [2016a].

In Table 4.7 we can see that 26 percent of the verbs are mistranslated and about

10 percent of the arguments. Mistranslated verbs are problematic since the feature produces the selectional association scores for the wrong verb. Although the semantic affinity is mutual, the formulation of the score conditions on the verb. In the cases when both the verb and the argument are mistranslated the association score might be high although the translation is not faithful to the source.

4.6.2 Evaluation of the Selectional Preference Feature

Next, we determine the effectiveness of our selectional association features. We compare the two different selectional association features described in Section 4.4.2: *SelAssoc_L* and *SelAssoc_C* . We report the results of automatic evaluation in Table 4.8.

Neither of the features improved the automatic evaluation scores. The *SelAssoc_L* suffers more from data sparsity than the *SelAssoc_C*, which in turn is overgeneralizing due to noisy clustering. Adding both features compensates for these issues, however we only see a slight improvement in BLEU scores for shorter sentences[6]: 25.59 compared to 25.40 for the baseline system. We further investigate whether sparse features are more informative.

We changed the format of the features in order to experiment with sparse features. By using sparse features we let the tuning algorithm discriminate between low and high values of the *SelAssoc* score. For each of the *SelAssoc* features we normalized the scores to have zero mean and standard deviation one and mapped them to their corresponding percentile. A sparse feature was created for each percentile, below and above the mean [7] resulting in a total of 20 sparse features (10 for *SelAssoc_C* and 10 for *SelAssoc_L*). However this formulation of the feature also did not improve the evaluation scores as shown in the fifth row of Table 4.8. We also tried a binned version of the *SigmaPMI*[8] measure of selectional preferences proposed by Kiperwasser and Goldberg [2015] for improving syntactic parsing. This measure also did not improve translation according to the BLEU score reported on the last row of Table 4.8.

The lack of variance in automatic evaluation scores can be explained by: a) the feature touches only a few words in the translation and b) the relation between a predicate and its argument is identified at later stages of the bottom-up chart-based decoding when many lexical choices have already been pruned out. The *SelAssoc* scores, similar

[6] 2,701 sentences with more than 5 words and at most 15 words

[7] Up to two standard deviations below the mean and three standard deviations above the mean.

[8] The sigmoid function applied to the point-wise mutual information (PMI) between the head words of the predicate and argument.

System	BLEU -c	HWCM
Baseline	26.45	24.47
+ SelAssoc_L	$26.41_{-.04}$	$24.52_{+.05}$
+ SelAssoc_C	$26.48_{+.03}$	$24.54_{+.07}$
+ SelAssoc_L + SelAssoc_C	$26.48_{+.03}$	$24.47_{+.00}$
+ Bin (SelAssoc_L + SelAssoc_C)	$26.37_{-.08}$	$24.53_{+.06}$
+ RDLM–P_w (1, 0, 0)	$26.35_{-.10}$	$24.75_{+.28}$
+ RDLM–P_w (2, 1, 1)	$26.38_{-.07}$	$24.83_{+.36}$
+ Bin(SigmaPMI)	$26.41_{-.04}$	–

Table 4.8: BLEU and head-word chain metric (HWCM) results for string-to-tree systems with the selectional preference (*SelAssoc*) and the relational dependency language model (RDLM–P_w) features. *SelAssoc_L* uses lemmas as word class representations, *SelAssoc_C* uses 500 word clusters. The triples in parentheses indicate the context size for ancestors, left siblings and right siblings respectively. The RDLM–P_w configuration (1, 0, 0) captures similar syntactic context as the selectional preference features. *Bin(·)* stands for the binned version of a feature representation. This table was adapted from Nădejde et al. [2016a].

to mutual information scores, are sensitive to outlier events with low frequencies in the training data. In the next section we investigate whether a more robust model would mitigate some of these issues and experiment with a neural relational dependency language model (RDLM) [Sennrich, 2015].

4.6.3 Comparison with a Relational Dependency LM

The RDLM [Sennrich, 2015] is a feed-forward neural network which learns two probability distributions conditioned on a large syntactic context described in Eq 4.3: P_w predicts the head word of the dependent and P_l the dependency relation. We compare our feature with RDLM–P_w.

For training the RDLM–P_w we use the parameters for the feed-forward neural network described in Sennrich [2015]: 150 dimensions for input layer, 750 dimensions for the hidden layer, a vocabulary of 500,000 words and 100 noise samples. We train

	SelAssoc_L		SelAssoc_C	
System	main	prep	main	prep
Baseline	0.067	0.039	0.164	0.147
+ SelAssoc_L + SelAssoc_C	0.074	0.041	0.175	0.305
Reference	0.077	0.043	0.186	0.163

Table 4.9: Average selectional association scores for the test set. Scores are aggregated over the *main* and *prep* argument types. *main* arguments include: nsubj, nsubjpass, dobj, iobj. This table was adapted from Nădejde et al. [2016a].

the RDLM–P_w on the target side of the parallel data. Although we use less data than for training the *SelAssoc* features, the neural network is inherently good at learning generalizations and selecting the appropriate conditioning context.

We experiment with different configurations for RDLM–P_w by varying the number of ancestors as well as left and right siblings:

- ancestors = 1, left = 0, right = 0
- ancestors = 2, left = 1, right = 1

The first configuration captures similar syntactic context as the *SelAssoc* features. The only exception is the *prep* relation for which the head of *pobj*, the actual preposition, is the first ancestor of the argument. Considering the example in Figure 4.2, the first ancestor for the noun *Prime* is the noun *Minister*, the second ancestor is the verb *met*, the sibling to the left is the determiner *the* and the sibling to the right is the preposition *of*. The results are shown in the last two lines of Table 4.8 and the configuration is marked between parentheses for the ancestors, left siblings and right siblings respectively.

The RDLM–P_w models achieve higher HWCM scores than the selectional preference feature, which is to be expected since the RDLM–P_w considers all dependency relations. However, there is not a significant contribution from having a larger syntactic context.

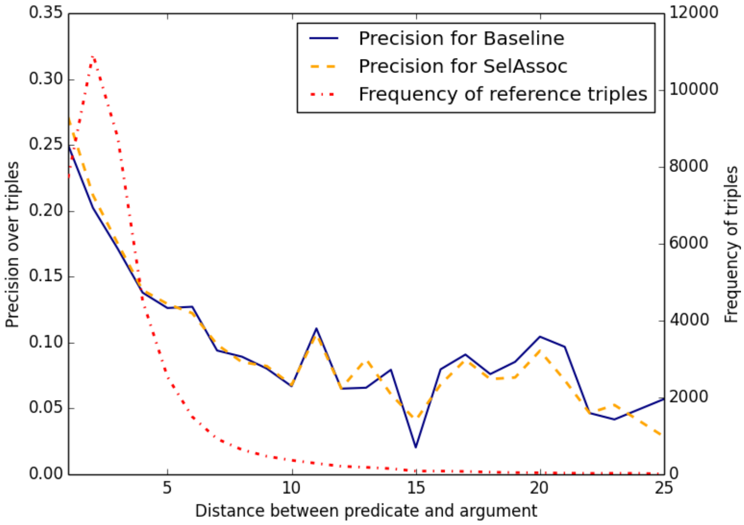


Figure 4.3: Frequency of triples (Y axis right) and translation precision of baseline as well as *SelAssoc* model (Y axis left) with respect to the distance between the predicate and its arguments, in tokens (X axis). This figure was adapted from Nădejde et al. [2016a].

4.6.4 Analysis

We now investigate possible reasons for the low impact of our selectional preference features. We look at how frequently our features are triggered, and how precision is influenced by the distance between predicates and their arguments.

Firstly, we are interested in how often the feature triggers and how it influences the overall selectional association score of the test set. On average, 4.85 triples can be extracted per sentence, from the syntactic tree produced by our system. Out of these, 4.35 triples get scored by the *SelAssoc_C* feature and 3.56 by the *SelAssoc_L* feature. The selectional association scores are higher on average for our system than for the baseline as shown in Table 4.9. To extract the triples for the baseline system, we parse the translated sentences with the Stanford parser. The *SelAssoc_C* feature seems to overgeneralize for the *prep* relations as the scores are on average higher than for the reference triples. We therefore conclude that our feature is having an impact on the translation system.

Secondly, we want to understand the interaction between the *SelAssoc* features and the language model. For this purpose we compute the frequency and translation precision of triples with respect to the distance between the predicate and its arguments.

Figure 4.3 shows the frequency of triples extracted from the reference sentence as well as the translation precision of triples extracted from the output of the translation systems. For more reliable precision scores, we lemmatized all predicates and arguments. 93% of the arguments are within a 5 word window from the predicate and therefore fall within the language model context. For these triples we see only a slight increase in precision for our system. This result indicates that for predicates and arguments that are close to each other, the feature is not adding much information. As the distance increases the precision decreases drastically for both systems: already at a distance of 5 words, the precision is down to 13%.

4.6.5 Discussion

One reason for the small impact of both the *SelAssoc* and the RDLM–P_w features is the poor quality of the syntactic trees produced by the decoder for longer sentences. In Section 3.4 we reported the results of the manual evaluation of the syntactic trees generated by a string-to-tree system, showing many attachment errors. As we argued in that section, we cannot automatically evaluate the quality of the syntactic trees generated by the string-to-tree system. However, the baseline precision scores reported in Figure 4.3 for longer distances are also an indicator of the quality of the syntactic trees. A larger distance between the predicate and argument also implies a more complex syntactic structure, for which the system will make more attachment mistakes. In such cases, when the wrong triples are extracted and precision scores are low, both *SelAssoc* and RDLM–P_w features will give confusing signals to the system. To confirm this, we propose in future work to re-evaluate the features on a subset of translated sentences exhibiting several attachment errors, which can be manually identifed.

In more complex sentences the features may score modifiers that are not important for disambiguating the verb. The example in Figure 4.4 has several prepositional modifiers but only *"on tour"* could help disambiguate the verb *"brachen auf (went)"*. In such cases identifying the semantic roles of the modifiers in the source and projecting them on the target might be useful for better estimation of semantic affinities.

The error analysis on short sentences showed that the mis-translation of verbs is a problem for the baseline system. This is confirmed by the low precision scores[9] for verb translation shown in Table 4.10. Although there is a slight improvement in precision, generally mistranslated verbs impact our features as the selectional association

[9] The precision scores were computed over verb lemmas extracted automatically from the test sets. In total 21,633 source verbs were evaluated.

Source	Das 16-jährige Mädchen und der 19-jährige Mann **brachen** kurz nach Sonntagmittag in Govetts Leap in Blackheath **zu ihrer Tour auf**.
Reference	The 16-year old girl and the 19-year old man **went on their tour** shortly after Sunday lunch at Govetts Leap in Blackheath.
Baseline	The 16-year old girl and the 19-year old man **broke** shortly after Sunday lunch in Govetts Leap in Blackheath **on their tour**.

Figure 4.4: Examples of a complex sentence with multiple prepositional modifiers. Information about semantic roles is needed to identify the relevant prepositional modifier. The incorrectly translated phrase is in bold.

System	Precision
baseline	46.10
+ SelAssoc_L + SelAssoc_C	$46.26_{+.16}$
+ RDLM–P_w (2, 1, 1)	$46.31_{+.21}$

Table 4.10: Evaluation results of verb translation on the test set. Precision scores are computed over verb lemmas against the reference translations. This table was adapted from Nădejde et al. [2016a].

score is computed for the wrong verb. In future work, we propose to conduct oracle experiments in which we force the system with the selectional association feature to choose the correct verb. Then, we can measure to which extend the feature scores and translation quality improved for the verbs that the baseline system mistranslated.

4.7 Conclusions

This chapter explores whether knowledge about semantic affinities between the target predicates and their argument fillers is useful for translating ambiguous predicates and arguments. We introduce three variants of a selectional preference feature for string-to-tree statistical machine translation based on the selectional association measure of Resnik [1996]. We compare our features with a variant of the neural relational dependency language model (RDLM) [Sennrich, 2015] and find that neither of the features improves automatic evaluation metrics. The impact analysis of the selectional preference feature indicates that for predicates and arguments that are close to each other, the feature is not adding much information. As the distance between predicates and arguments increases, we observe that the system performance degrades. We propose as future work, to measure to what extent does the quality of the syntactic trees generated by the system impact the selectional preference feature. Furthermore, it is not possible to accurately model semantic affinities between verbal predicates and their arguments if the verbs are often mistranslated. In the next chapter we analyse in more depth the problem of mistranslated verbs and propose a Neural Verb Lexicon Model conditioned on a wide source-side syntactic context.

Chapter 5

A Neural Verb Lexicon Model

5.1 Introduction

In the previous chapter, we explored modeling semantic affinities between the target predicates and their argument fillers. Our analysis showed that as the distance between a predicate and its argument grows, and as the target syntactic structure becomes more complex, the translation quality of the pair drops. Furthermore, verbs are often mistranslated which negatively impacts the proposed Selectional Preferences model. These analyses motivate our work on improving lexical choice for verbs using source-side sentence-level context, presented in this chapter.

String-to-tree MT systems may translate verbs without lexical or syntactic context on the source side and with limited target-side context. As we show in this chapter, the lack of context is one reason why verb translation recall is as low as 45.5%[1]. We propose a Neural Verb Lexicon Model (NVLM) which addresses specifically the problem of verb translation in string-to-tree systems by looking at the source-side syntactic context. We train a verb-specific lexicon model since verbs have the most outgoing dependency relations, are central to semantic structures and therefore would benefit most from the source-side syntactic context.

Several Discriminative Word Lexicon (DWL) models with source-side features have addressed the problem of word sense disambiguation in phrase-based MT [Mauser et al., 2009, Niehues and Waibel, 2013, Herrmann et al., 2015]. Our proposed verb lexicon model is trained with a feed-forward neural network (FFNN) which, unlike DWL models, allows parameter sharing across target words and avoids exploding feature spaces. Previous lexicon models trained with FFNN [Ha et al., 2014] and using con-

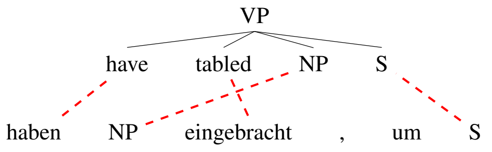
[1] Recall is computed against the reference translations.

text extracted from the source sentence were inefficient to train and did not scale to large vocabularies. We avoid scaling problems by choosing the context which is most relevant for verb prediction in a pre-processing step, from the source-side dependency tree.

This chapter is structured as follows. In Section 5.2 we exemplify why verb translation is problematic for string-to-tree systems and contrast our proposed model with prior work on discriminative word lexicon and rule selection models. In Section 5.3 we present a thorough analysis of verb translation conducted on a German→English string-to-tree system, and we determine to what extent this is a problem for the state-of-the-art system. Section 5.4 describes our proposed neural verb lexicon model and presents ablation experiments evaluated in terms of verb prediction accuracy. Finally, in Section 5.5, we investigate whether the verb lexicon model is able to improve translation quality by integrating the model as an additional feature for re-ranking the output of the string-to-tree system.

5.2 Background

String-to-tree MT systems handle long distance reordering with synchronous translation rules. In Figure 5.1, we repeat the example from Figure 1.2 of a German-English synchronous CFG rule translating a verb phrase, which contains the lexical items corresponding to the verbs *"haben eingebracht"* and two non-terminals corresponding to the main verb's arguments.



$$\text{VP} \rightarrow \text{have tabled NP}_0\ \text{S}_1\ |||\ \text{haben X}_0\ \text{eingebracht um X}_1$$

Figure 5.1: Reordering translation rule. The target syntactic sub-tree and the alignment of the non-terminals to the source-side spans are depicted at the top. The corresponding synchronous context-free grammar rule is depicted at the bottom.

This synchronous CFG rule reorders the verb and its arguments according to the target side word order. To allow the reordering of the NP (noun phrase) and S (sentence) constituents, they will be translated by independent rules and therefore the verb will be translated without lexical context. In the next section, we show that 20% of the main verbs are translated by a lexical rule which is the equivalent of a one word phrase-pair. The language model context is also limited, and will capture at most the verb and one main argument. Due to the lack of a larger source or target context the verb is often mistranslated. In Figure 5.1, the verb *"eingebracht"* is translated as *"tabled"*, which in American English means *"to postpone consideration of"*, while in British English it means *"to propose"*, which is also the meaning conveyed by the reference translation. This rule is shown in context in Figure 5.4, Section 5.5, where we describe the example in detail. We comment on how our proposed model chooses a translation that is accurate given the context and not ambiguous in either British or American English.

In this chapter we propose to improve lexical choices for verbs by learning a verb-specific lexicon model conditioned on context extracted from the syntactic structure of the source sentence. We train a Neural Verb Lexicon Model with a feed-forward neural network (FFNN) and select the relevant context of the source verb following its dependency relations.

Several approaches have been proposed to improve word sense disambiguation (WSD) for machine translation by integrating a wider source context than is available in typical translation units. For phrase-based MT, one such approach is to learn a discriminative lexicon model as a maximum-entropy classifier which predicts the target word or phrase conditioned on a highly dimensional set of sparse source-side features. Carpuat and Wu [2007] train a classifier for each source phrase and use features engineered for Chinese WSD to choose among available phrase translations. Tamchyna et al. [2016] propose a similar model that uses target-side features and that shares parameters across all source phrases. Mauser et al. [2009] introduced the Discriminative Word Lexicon (DWL) which models target word selection independently of which phrases are used by the MT model. The DWL is a binary classifier that predicts whether a target word should be included or not in the translation, conditioned on the set of source words. Niehues and Waibel [2013] extend the DWL with target-side context and bag-of-n-gram features aimed at capturing the structure of the source sentence. Herrmann et al. [2015] extend the work of Niehues and Waibel [2013] with other source-side structural features such as dependency relations.

For syntax-based MT, discriminative models have been used to improve rule selection [Braune et al., 2015, 2016, Liu et al., 2008]. Rule selection involves choosing the correct target side of a synchronous rule given a source side and other features such as the shape of the rule and the syntactic structure of the source span covered by the rule.

Braune et al. [2016] proposes a global discriminative rule selection model for hierarchical MT which allows feature sharing across all rules and which incorporates a wider source context such as words surrounding the source span. However, the model only disambiguates between rules with the same source side. Considering that hierarchical rule tables are much larger than phrase tables, the discriminative rule selection models are much more computationally demanding than the discriminative lexicon models.

The aforementioned DWL models train a separate classifier for each target word or phrase. The classifier parameters are not shared across target words and the feature combinations are not learned but generated through cross-products of feature templates. Joint translation models trained with feed forward neural networks (FFNN) [Devlin et al., 2014] address these problems, however, these are efficiently trained only on local context. Ha et al. [2014] proposes a joint model with sentence-level context similar to the DWL but trained with FFNN. However, the resulting network is very large and inefficient to train and therefore the model does not scale to large vocabularies.

Our work is similar to Herrmann et al. [2015] as we select relevant source context following the dependency relations between the verb and its arguments. However, we take advantage of parameter sharing and avoid the problem of exploding feature space by training our model with a FFNN. Different from Ha et al. [2014], we are able to incorporate sentence-level context by taking advantage of the syntactic structure of the source sentence. We train a verb specific lexicon model with the knowledge that verbs have the most outgoing dependency relations, are central to semantic structures and therefore would benefit most from a source-side syntactic context. We train a lexicon model and not a rule selection model as we are trying to address the problem of lexical translation of verbs in string-to-tree systems. Moreover, by predicting only the target verb we can simplify the prediction task and train a smaller model.

5.3 Verb Translation Analysis

In this section, we present an analysis of verb translation in syntax-based models for the German to English language pair. We estimate the impact of a verb lexicon model through the percentage of verbs that would benefit from source-side context and determine the extent to which we could reduce the number of verbs lost in translation by re-ranking the n-best list.

The string-to-tree system used for this analysis is trained on all available data from

WMT15 [Bojar et al., 2015] and is described in more detail in Section 5.5. The evaluation test set consists of newstest2013, newstest2014 and newstest2015 totaling 8,172 sentences. To identify corresponding source and target verbs for this analysis, the source side of the parallel data is parsed with dependency relations using ParZU [Sennrich et al., 2013] and the target side is tagged with part-of-speech labels using TreeTagger [Schmid, 1994].

Firstly, we present in Table 5.1 a breakdown of counts at token level for verbs identified in the source sentences. Verbs were first identified by their part-of-speech label and then the dependency relations were used to distinguish between auxiliary verbs (except modals) and main verbs. Main verbs represent 73.2% of all verbs while only 20.0% are auxiliary verbs. The other 6.8% of words labeled as verbs are either modals or can not be identified as either auxiliaries or main verbs.

	count	percentage
source verbs	23,492	100.0
|__ auxiliary verbs	4,689	20.0
|__ misaligned verbs	934	3.9
|__ main verbs	17,210	73.2
|__ particle verbs	1,589	6.7
|__ **target verbs**	**11,161**	**47.5**
|__ misaligned verbs	2,850	12.1
|__ modals + other	1,593	6.8
|__ lexical rules	4,905	20.8

Table 5.1: Breakdown of source verb categories in newstest2013-2015. Token level counts. Our analysis focuses on main verbs that can be aligned with target verbs, highlighted in bold. This table was adapted from Nădejde et al. [2016b].

The first problem for verb translation is that in the automatic word alignment stage, prior to GHKM rule extraction, some verbs are aligned with at least one comma or not aligned at all, which breaks the constraints of rule extraction. Because of such misalignments, the grammar will contain rules that translate the verb with a comma or drop the verb on the target side. We measured the degree of misalignment found in the GIZA++ automatic word alignment of the test set sentences. A total of 16% of verbs are misaligned, out of which 3.9% are auxiliaries[2] and 12.1% are main verbs.

[2] Not all German auxiliaries need to be translated into English, since different forms of past tense can

Source:	Ich **gehe** heute abend **aus**.
Target:	I'm going out tonight.
Source:	Letztes Jahr **ging** ich **fort**.
Target:	Last year I left.

Figure 5.2: Example of German verbs with separable particles (verb and particle in bold).

In this work we will focus on translation of main verbs as they carry the semantic information. To evaluate a verb lexicon model more accurately, we focus only on source verbs that align with target verbs, thus avoiding the misalignment problem. We identify the corresponding source and target verb pairs from newstest2013-2015 using the word alignment and the part-of-speech labels, and obtain a total of 11,161 verb occurrences which we use for evaluation.

A second problem for verb translation is that synchronous rules may translate the verb independently of its arguments. Table 5.1 shows that 20.8% of the verbs are translated with lexical rules, i.e rules such as *VBD* → *tabled* ||| *eingebracht*. Lexical rules are the equivalent of one-word phrase-pairs in phrase-based SMT and as such, they do not provide any syntactic context. When translating verbs with lexical rules the system relies only on language model context to disambiguate the verb. However, the language model context might become available only in later stages of bottom-up chart-based decoding, when larger synchronous rules are applied to connect and reorder the verb and its arguments. To address this problem we propose a verb lexicon model that uses a wide source-side context to predict the target verb.

An interesting class of German verbs are those with separable particles which are moved at the end of the sentences for present tense or imperative. For example the verbs *ausgehen (to go out)* and *fortgehen (to leave)* have the root *gehen (to walk)*. However, the particles *aus* and *fort* separate from the root and change its meaning, which leads to a specific type of translation error. We give example sentences below.

We continue to evaluate the tree-to-string system in terms of verb translation recall. The translation recall shown in Table 5.2 is computed over the 11,161 instances of main source verbs which were aligned with a reference verb using GIZA++ word alignment. To compute recall, we count how many of these 11,161 reference verbs were correctly translated by the system. We report separate recall numbers for the

be used. For example, *habe gegessen* translates as *ate*.

correct verb translations found either in the 1-best hypothesis, among all the n-best hypotheses or among all entries in the rule table that can translate the source verb.

source	token	lemma
1-best	45.54	53.14
1000-best	72.87	79.24
rule table	91.85	-

Table 5.2: Verb translation recall for 1-best translation, 1000-best lists and rule table computed over verbs from newstest2013-2015. Measured separately at token and lemma level. This table was adapted from Nădejde et al. [2016b].

Verb translation recall is only 45.54% at token level for the 1-best output of the syntax-based system. However verb recall in the 1000-best list is much higher, at 72.87%. This result indicates that better translation options are available and re-scoring these options could result in improved 1-best verb translation recall. Furthermore, by looking at the target side of all the verb translations in the rule table we can see that the reference translation is available in almost 92% of the cases.

Finally, we compare the reference translations and the system translations of the verbs in terms of their rank among all translation candidates. For this purpose we order the translation options for each of the source verbs according to the direct translation probability $p(target|source)$. For each source verb, we compute the rank of the corresponding verb translation found in the reference and the rank of the verb translation produced by the syntax-based system. We can see in Table 5.3 that the reference translations have rank 1 only 50.71% of the cases compared to 65.48% for the system translations. Since the reference translation of the verb is often less probable than the selected one, we are dealing with modeling errors. Re-scoring only the top 10 translation options could improve the translation model accuracy from 50.7% to 68.2%. [3]

5.4 Verb Lexicon Model

In the previous section we have shown that string-to-tree MT systems translate verbs with low recall and accuracy. Better translation candidates can be found in the 1000-

[3] The accuracy of the translation model and the percentage of reference verbs that are ranked 1st may be underestimated because we have only one reference translation available.

source	rank = 1st	rank < 5th	rank < 10th
reference	50.71	56.30	68.25
system	65.48	73.90	84.87

Table 5.3: Percentage of verb translations that are ranked first or higher than 5th or 10th rank in the rule table. We report results for the translations of the source verbs from newstest2013-2015 found in the reference or 1-best system output. This table was adapted from Nădejde et al. [2016b].

best lists. However, at least 20% of verbs are scored without contextual information by the translation model.

In this section we propose a verb lexicon model that uses source side context to predict the target verb. Both the source word sequences and the source syntactic structure are readily available at early stages of decoding. In contrast, target side context for verbs, such as their arguments, becomes available at later stages of decoding when larger synchronous rules are applied. Moreover, the target syntactic structure generated during decoding is not sufficiently accurate for extracting arguments of the target verb. While similar lexicon models have been proposed in the literature [Mauser et al., 2009, Niehues and Waibel, 2013, Herrmann et al., 2015], this work explores whether a source syntactic context is more informative for predicting target verbs than a window context centered on the source verb. We propose a verb specific model since verbs have more arguments and longer syntactic dependencies than other words and therefore would benefit from a wider source-side context. Our verb lexicon model is a feed-forward neural network trained with the NPLM toolkit [Vaswani et al., 2013].

We first show that verbs are harder to predict cross-lingually than other words for the German-English language pair. For this purpose we train a generic lexicon model that takes as input a 5-word window centered on the source word of interest and outputs the corresponding target word. The generic model is trained on all words from WMT15 parallel data and evaluated on either all words from newstest2013 - 2015 test sets or on the subset of 11,161 main verbs selected as described in the previous section. Table 5.4 shows that the generic lexicon model performs worse at predicting target verbs: perplexity is higher, 26.20 for verbs compared to 23.62 for all words, and accuracy is lower, 43.67 for verbs compared to 50.62 for all words. This reinforces our argument that we need a verb specific lexicon model.

	perplexity	acc@1	acc@5	acc@15
all words	23.62	50.62	70.51	78.47
verbs only	26.20	43.67	67.88	78.69

Table 5.4: Perplexity and accuracy (acc; at ranks 1, 5 and 15) of the generic lexicon model reported over all words and over verbs only, on newstest2013-2015. This table was adapted from Nădejde et al. [2016b].

5.4.1 Syntactic Context

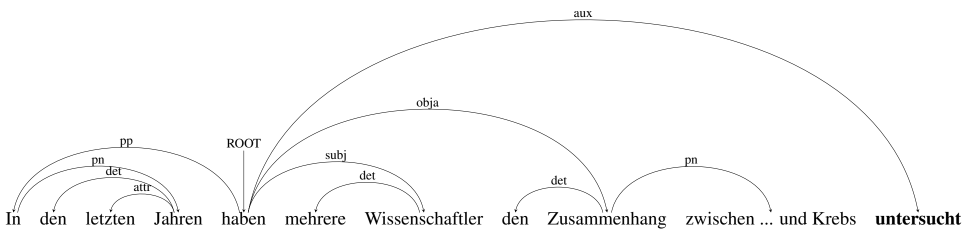
In order to predict target verbs more accurately, as well as to train the models more efficiently, we learn a specialized verb lexicon model in the form of a feed-forward neural network. The network receives a fixed number of input tokens extracted from the source sentence and predicts a target verb.

Next, we explore whether a source-side syntactic context is more informative for predicting the target verb than a window context. Since the syntactic context is extracted from the source sentence we can include most of the verb's dependents, in particular the core arguments that carry most semantic information relevant to verb disambiguation. We also provide the verb lexicon model with a feature encoding the subcategorization frame, as this information was useful for inducing verb classes in a monolingual setting [Sun and Korhonen, 2009, Schulte im Walde, 2006].

From the dependency parse of the source sentences we extract the following syntactic context: the parent of the main verb, the first prepositional modifier[4] and its preposition, up to three other dependents and the verb particle, if any. If an auxiliary verb is present, we attach all its dependents[5] to the main verb and leave the auxiliary as the parent of the main verb only. We then create a subcategorization token by concatenating the dependency relations of all verb dependents. In order to reduce sparsity of the data we add the lemma of each word in the syntactic context. If all types of syntactic context are considered, including the lemma factors, the network will receive 16 input tokens. We show an example of source syntactic context for a verb in Fig-

[4] Preliminary experiments did not show improvements when considering more prepositional modifiers. Furthermore, when the verb does not have a second prepositional modifier, two $< null >$ tokens are required as input to the network instead. When we considered a larger syntactic context (more dependents or prepositional modifiers) and the number of $< null >$ tokens in the input increased, we faced problems training the neural networks.

[5] Dependents are identified by following outgoing edges.


aux
obja
pp
ROOT
pn
subj
pn
det
det
det
attr
det
In den letzten Jahren haben mehrere Wissenschaftler den Zusammenhang zwischen ... und Krebs **untersucht**

Reference:	In recent years , a number of scientists have **studied** the links [between ... and cancer]					
Window context	und Krebs **untersucht** $< /s > < /s >$					
Syntactic context	source verb	parent	dependents	pp modifier	subcat	particle
word:	**untersucht**	haben	Wissenschaftler Zusammenhang <null>	in Jahren	pp_subj_obja	<null>
lemma:	untersuchen	haben	Wissenschaftler Zusammenhang <null>	in Jahr	-	-

Figure 5.3: Example of a source-side dependency graph, as well as the window and syntactic context that were extracted from it for the source verb untersucht (studied; in bold). With both word and lemma factors, and counting the $< null >$ tokens, the network receives as input a total of 16 tokens for the syntactic context.

ure 5.3. In this example there are 9 pieces of context[6], out of which 7 have both a word and lemma factor, resulting in a total of 16 inputs for the neural network. The subcategorization feature (subcat) for the verb "untersucht" is *pp_subj_obja*, and does not contain the label *aux* since the auxiliary is the parent of the main verb and not a dependent. In German, nouns and their articles are inflected differently depending on the case and therefore the label *obja* marks that the object is in the accusative case.

5.4.2 Experimental Setup and Evaluation

We train the models with the NPLM toolkit [Vaswani et al., 2013] implementing a feed-forward neural network and choose model hyper-parameters following Sennrich [2015]. We use 200 dimensions both for the input embeddings and for the single hidden layer. Both the input and output vocabularies consist of the 500,000 most frequent words (including named entities, foreign words and numbers which may appear frequently in a noisy parallel corpus). The input vocabulary is shared for words and lemmas. When adding the subcategorization feature (subcat) we increase the input vocabulary by 80,000. We use the "rectifier" activation function, a batch size of 256, and train for at most 25 iterations.

[6] We count the $< null >$ tokens because the neural verb lexicon model is trained with a feed-forward neural network which requires a fixed number of inputs.

We train the models on all the parallel training data available at WMT15 and we use a development set of 2,000 sentences for early stopping of training. The models are evaluated in terms of perplexity and accuracy over the verbs extracted from newstest2013, newstest2014, newstest2015.[7] The data is described in Table 5.5. The source side of the parallel data is parsed with dependency relations using ParZU [Sennrich et al., 2013] and the target side is tagged with part-of-speech labels using TreeTagger [Schmid, 1994].

	Train	Tune	Test
sentences	4,472,694	2,000	8,172
verb tokens	5,945,637	2,419	11,211

Table 5.5: Number of sentences as well as verb tokens in the training, tuning and test sets used for the Neural Verb Lexicon Model experiments.additional blah. should mention somewhere how verbs were counted. e.g. which POS tags count as verbs

Table 5.6 shows the performance of different models. The accuracy of the verb lexicon model trained with a 5-word window context is 50.57%, compared to 43.67% the accuracy of the generic lexicon model reported on the last row and in Table 5.4. This result shows that training a verb-specific model is beneficial. In Table 5.3 we showed that the direct translation probability predicts the correct translation for 50.71% of the verbs that have a translation in the rule table. The prediction of the verb lexicon model with window context matches the reference translation in 50.57% of the cases, however its top 5 accuracy is 76.27% compared to only 56.30% for the direct translation probability.

Increasing the window context size to 7 words does not improve performance of the verb lexicon model. In contrast, providing a syntactic context of similar size as input to the network results in a lower perplexity and higher accuracy. Adding the lemma factor helps for both types of context in terms of perplexity, however the accuracy is higher only for the syntactic context. The subcategorization feature did not improve accuracy, which we attribute to the fact that the other syntactic context already provides a strong signal about the presence of the main arguments and the prepositional modifier. Furthermore, SCFG rules translating verbs partially encode information about the

[7] In Section 5.3 we compared reference and system translations of 11,161 source verbs. The remaining verbs up to 11,211 were discarded since there was no corresponding system translation in the 1-best output.

context	factors	size	perplexity	acc@1	acc@5	acc@15
window	word	5	**27.81**	**50.57**	76.27	85.04
window	word	7	27.98	50.57	75.55	85.03
window	word, lemma	10	27.20	50.54	75.90	85.42
syntactic	word	7	26.49	51.21	76.26	85.36
syntactic	word, lemma	14	24.99	51.46	77.12	85.83
syntactic	word, lemma, subcat	15	25.16	51.54	76.83	85.82
syntactic	word, lemma, subcat, particles	16	**24.84**	**51.99**	77.54	85.96
baseline	word	5	26.20	43.67	67.88	78.69

Table 5.6: Evaluation of different configurations of the verb lexicon model according to perplexity and accuracy (acc) at ranks 1, 5 and 15. The size column indicates the number of inputs to the neural network. The token level verb prediction accuracy is reported over newstest2013-2015. The baseline is the generic lexicon model. The configurations with the lowest perplexity and highest acc@1 scores respectively are in bold. This table was adapted from Nădejde et al. [2016b].

subcategorization frame (except when the verb is translated with a lexical rule). For example, the SCFG rule in Figure 5.1 encodes that the verb expects at least one *NP* argument. Finally, adding the particle as separate input increases the accuracy leading to a total improvement of 1.5% over the baseline window context.

In the next section we investigate whether the verb lexicon model is able to improve translation quality by integrating the model as an additional feature for re-ranking machine translation output. As we showed in the verb translation analysis, re-ranking the 1000-best list could potentially improve verb recall from 45% to 72%.

5.5 Machine Translation Evaluation

Our baseline system for translating German into English is the Moses string-to-tree toolkit implementing GHKM rule extraction [Galley et al., 2004b, 2006b, Williams and Koehn, 2012]. The rule extraction parameters and the setup of the system were previously described in Section 3.3.2 and in Nădejde et al. [2013], Williams et al. [2014]. We train the system on all available data provided at WMT15[8] [Bojar et al., 2015]. We report the number of sentences in the training and tuning sets in Table 5.5.

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

		BLEU		METEOR	HWCM
context	factors	dev	test	test	test
Baseline	-	$26.18_{\pm 0.0}$	$26.10_{\pm 0.0}$	$29.95_{\pm 0.0}$	$25.27_{\pm 0.0}$
+ window	word	$-0.13_{\pm 0.08}$	$-0.39_{\pm 0.26}$	$-0.13_{\pm 0.14}$	$-0.20_{\pm 0.20}$
+ dependency	word, lemma, subcat	$-0.06_{\pm 0.05}$	$-0.22_{\pm 0.12}$	$-0.07_{\pm 0.08}$	$-0.10_{\pm 0.09}$
+ dependency	word, lemma, subcat, particles	$-0.13_{\pm 0.06}$	$-0.37_{\pm 0.19}$	$-0.14_{\pm 0.06}$	$-0.19_{\pm 0.18}$

Table 5.7: Results of re-ranking the 1000-best list of a baseline string-to-tree system with different configurations of the verb lexicon model as an additional feature. BLEU , METEOR and HWCM scores are reported over newstest2015 (2,169 sentences and 3,002 reference verbs) with standard deviation shown from 3 runs of minimum error rate training (MERT). This table was adapted from Nădejde et al. [2016b].

At decoding time we give a high penalty to glue rules and allow non-terminals to span a maximum of 50 words. We report evaluation scores over the newstest2015 data set (2169 sentences, 3002 verbs).

We integrate the verb lexicon model in re-ranking by adding two new features scores in addition to the baseline features:

- A counter for the source verbs translated by the n-best hypothesis.
- Verb lexicon model scores aggregated over all main verbs.[9]

The weights for the new feature scores and for the baseline features are re-tuned using MERT on the tuning set. We run MERT three times and for each set of weights we re-ranked the machine translation output.

Table 5.7 shows average cased BLEU [Papineni et al., 2002], METEOR [Lavie and Denkowski, 2009] and HWCM scores, as well as the standard deviation for the three different tuning runs. When adding the verb lexicon model there is a small decrease in evaluation metrics' scores: less than 0.4% for BLEU and less than 0.2% for METEOR and HWCM .

Table 5.8 shows average precision, recall and F1 scores for verb translation, as well as the standard deviation for the three different tuning runs. To compute recall we divide the number of reference verbs correctly translated by the 1-best system hypothesis by the total number of verbs present in the reference translation. To compute

[9] By main verbs, we refer to the main source verbs that were translated to verbs, as identified with the word alignment reported by the SMT system.

		Precision		Recall		F1	
context	factors	token	lemma	token	lemma	token	lemma
Baseline	-	$56.91_{\pm 0.0}$	$65.18_{\pm 0.0}$	$47.86_{\pm 0.0}$	$54.83_{\pm 0.0}$	$51.99_{\pm 0.0}$	$59.56_{\pm 0.0}$
+ window	word	$+1.95_{\pm 0.66}$	$+2.04_{\pm 0.31}$	$+7.45_{\pm 0.42}$	$+8.34_{\pm 0.72}$	$+5.04_{\pm 0.32}$	$+5.57_{\pm 0.28}$
+ dependency	word, lemma, subcat	$+2.44_{\pm 0.80}$	$+2.39_{\pm 0.68}$	$+7.14_{\pm 0.76}$	$+7.8_{\pm 1.08}$	$+5.09_{\pm 0.09}$	$+5.42_{\pm 0.28}$
+ dependency	word, lemma, subcat, particles	$+2.70_{\pm 0.89}$	$+2.5_{\pm 0.72}$	$+7.36_{\pm 0.40}$	$+7.76_{\pm 0.06}$	$+5.34_{\pm 0.56}$	$+5.53_{\pm 0.32}$

Table 5.8: Results of re-ranking the 1000-best list of a baseline string-to-tree system with different configurations of the verb lexicon model as an additional feature. Precision, recall and F1 scores for verb translation are reported over newstest2015 (2,169 sentences and 3,002 reference verbs) with standard deviation shown from 3 runs of minimum error rate training (MERT). This table was adapted from Nădejde et al. [2016b].

precision, we divide the number of verbs present in both the reference and the 1-best system hypothesis by the total number of verbs present in this hypothesis.

On average the verb lexicon model improves precision up to 2.7%, recall up to 7.4% and F1 scores up to 5.3% at token level. The models with syntactic context improve precision more so than the models with window context, but not recall. This result motivates future work on analyzing how verb recall is affected by tuning feature weights towards BLEU , a precision based metric. We consider a 7% gain in verb translation recall to be more important than the small decrease in the evaluation metrics' scores since verbs are key pieces in semantic structures. Perhaps an even stronger verb lexicon model is needed in order to out-weight choices that only improve fluency. As future work, we propose to explore improving model coverage by making predictions for predicative nouns and improving model accuracy by conditioning on target context. Based on our analysis in Section 5.3, choosing from the n-best list allows for significant verb recall improvements, however this improvement may come at a cost to BLEU
.

In Figure 5.4 we give examples of correct verb translations produced by re-ranking the 1000-best list with the verb lexicon model.

In example a), the verb *eingebracht* is translated as *tabled* by the baseline system. This translation is ambiguous because it carries almost opposite meaning in American ("to postpone consideration of") and British English ("to propose"). On the last row of the example we show the synchronous translation rule used by the baseline system to translate the verb *eingebracht*. The rule correctly re-orders the noun-phrase *a bill* and the verb, as English objects should come after the verb. However the lexical choice

a)	Source	Die Kongress Abgeordneten haben einen Gesetzesvorschlag **eingebracht** , um die Organisation von Gewerkschaften als Bürgerrecht zu etablieren .								
	Reference	Congressmen have **proposed** legislation to protect union organizing as a civil right .								
	Baseline	Congressmen have **tabled** a bill to establish the organization of trade unions as a civil right .								
	Verb Lexicon	Congressmen have **introduced** a bill to establish the organization of trade unions as a civil right .								
	Syntactic context	source verb	parent	dependents			pp modifier		subcat	particle
	word:	**eingebracht**	haben	Kongress	Gesetzesvorschlag	etablieren	<null>	<null>	subj_obja_neb	<null>
	Translation rule	$VP \rightarrow \langle haben\ NP\ eingebracht\ um\ S\ ,\ have\ tabled\ NP\ S \rangle$								

b)	Source	die Ankläger **legten** am Freitag dem Büro des Staatsanwaltes von Mallorca Beweise für Erpressungen durch Polizisten und Angestellte der Stadt Calvia **vor** .								
	Reference	the claimants **presented** proof of extortion by policemen and Calvia Town Hall civil servants at Mallorca's public prosecutor's office on Friday .								
	Baseline	the prosecutor **went** to the office of the prosecutor of Mallorca Calvia evidence of extortion by police officers and employees of the city on Friday .								
	Verb Lexicon	the prosecutor **presented** evidence of extortion by police officers and employees of the city on Friday the office of the prosecutor of Mallorca Calvia before .								
	Syntactic context	source verb	parent	dependents			pp modifier		subcat	particle
	word:	**legten**	<null>	Ankläger	Büro	Staatsanwaltes	am	Freitag	subj_pp_objd_obja_pp_pp_avz	vor
	Translation rule	$VP \rightarrow \langle legten\ \hat{VP}\ ,\ went\ \hat{VP} \rangle$ $PP \rightarrow \langle NP\ vor\ ,\ to\ NP \rangle$								

Figure 5.4: Examples of correct verb translation produced by re-ranking the 1000-best list with the verb lexicon model. Main verb is marked in bold.

for the verb is made without knowledge of the lexical head of the object. The re-ranked translation *introduced* is both accurate and non-ambiguous in the context. The verb lexicon model prefers this translation because the words *Kongress* and *etablieren* appear in the syntactic context.

In example b), the verb *vorlegten (presented)* is translated incorrectly by the baseline system as *went*. This happens because the verb has a separable particle *vor* which is moved at the end of the sentence. The string-to-tree system is not able to find a rule that would make such a long distance reordering. Instead it translates the verb with two rules that are disconnected. The first rule translates the verb without any other context. The second rule incorrectly attaches the verb particle as a preposition to a noun-phrase. The verb lexicon model is able to produce the correct translation *presented* as the particle *vor* appears in the syntactic context.

In Figure 5.5 we give examples where the translations produced by re-ranking the 1000-best list with the verb lexicon model are worse than the 1-best translations.

In example a) the verb *geht weiter* is correctly translated by the baseline system as *goes on* but incorrectly translated by the verb lexicon model as *is*. The parser is not able to identify *weiter* as dependent of the source verb, therefore the verb lexicon model has limited context and gives a lower score to *goes* and a higher score to *is*. The

a)	Source	Und so **geht** das Leben , anders als das vieler anderer , für uns **weiter** .					
	Reference	So life **goes on** for us unlike for so many .					
	Baseline	And , unlike many others , life **goes on** for us .					
	Verb Lexicon	And so **is** the life , unlike many others , for us .					
	Syntactic context	source verb	parent	dependents	pp modifier	subcat	particle
	word:	**geht**	<null>	und so Leben	<null> <null>	koord_adv_subj	<null>
	Translation rule	$VBZ \rightarrow \langle geht\ ,\ goes\ \rangle$ $\hat{VP} \rightarrow \langle PP\ weiter\ ,\ on\ PP \rangle$					
b)	Source	Webster wird darüber hinaus **vorgeworfen** , am 4. Mai 2014 eine zweite Frau im Golf View Hotel in Naim im schottischen Hochland angegriffen zu haben .					
	Reference	Webster is then **charged** with attacking a second woman at the Golf View Hotel in Nairn in the Highlands on May 4 , 2014 .					
	Baseline	Webster is also **alleged** to have attacked a second woman in Naim's Golf View Hotel in the Scottish Highlands on 4 May 2014 .					
	Verb Lexicon	Webster is also **accused** of being a second wife in the Golf View Hotel on 4 May 2014 in Naim attacked in the Scottish Highlands .					
	Syntactic context	source verb	parent	dependents	pp modifier	subcat	particle
	word:	**vorgeworfen**	wird	Webster haben <null>	darüber hinaus	objd_pp_subjc	<null>
	Translation rule	$VBN \rightarrow \langle vorgeworfen\ ,\ alleged\ \rangle$ $VP \rightarrow \langle VBN\ ,\ VP\ zu\ haben\ ,\ VBN\ to\ have\ VP \rangle$					

Figure 5.5: Examples of translations produced by re-ranking the 1000-best list with the verb lexicon model that are worse than the 1-best translations. Main verb in bold.

wrong choice for the verb causes the resulting translation to have worse word order.

In example b) the verb *vorgeworfen* is incorrectly translated by the baseline system as *alleged*. The verb lexicon model is able to produce a better translation *accused*, however this affects the choice of other translation rules. As a result the second verb *angegriffen (attacked)* and its prepositional modifiers are incorrectly reordered in the translation. Future work should investigate whether the string-to-tree system can learn not to generate such fluency errors if the NVLM is integrated as a feature in the decoder. In this case the results should show improvements in both verb recall and BLEU scores.

5.6 Conclusions

In this chapter, we propose a verb lexicon model to improve the lexical choice for verbs in string-to-tree MT systems. We train the model with a feed-forward neural network that predicts the target verb conditioned on a wide source-side context. In Section 5.4 we show that a syntactic context extracted from the dependency structure of the source sentence improves model accuracy by 1.5% over the baseline window context.

In Section 5.5, we evaluate the verb lexicon model as an extra feature for re-ranking the output of a baseline string-to-tree MT system. The model improves verb translation precision by up to 2.7% and recall by up to 7.4% at the cost of a small (less than 0.5%) decrease in BLEU score. The verb lexicon model trained on the syntactic context improves verb translation precision more than the model trained on the window context, however recall is improved to the same extent.

In future work, we will explore whether we can further improve recall by tuning the feature weights using a metric that also rewards recall, such as METEOR and not just precision, as is the case with the BLEU metric. We also propose to consider predicative nouns as a means to improve model coverage and to provide the model with additional target-side context to improve precision. Another direction for future work could be to integrate the model as a feature in the string-to-tree decoder and investigate if this prevents errors appearing in other parts of the sentence when verb translation is improved.

Although in this work we improved some aspects of the translation, the strong independence assumptions made by string-to-tree SMT systems are causing syntactic and semantic translation errors which cannot be resolved by a linear combination of weak independent models. For this reason, we turn our attention to neural machine translation (NMT), an end-to-end machine learning framework, which considers the entire source sentence and target history as context when predicting the next target word. In the next chapter we show that, although NMT models are able to partially learn syntactic information from sequential lexical information, explicit target syntax can still improve translation quality.

Chapter 6

Syntax-aware Neural Machine Translation Using CCG

6.1 Introduction

In the previous chapters we augmented string-to-tree SMT systems with global source and target syntactic context to improve lexical consistency. While we improved this for verbs, the proposed models only partially address the strong independence assumptions made by SMT systems.

Over the last couple of years, neural machine translation (NMT) models showed significant improvement over strong SMT systems on many language pairs, including German→English. These models use the entire source context and target history when generating a translation, which is desirable when trying to learn long-distance dependencies and re-ordering. Even though NMT has strong learning capabilities, it has been shown that incorporating explicit source-side linguistic features can still improve translation quality [Luong et al., 2016, Sennrich and Haddow, 2016]. In this chapter, we examine the benefit of incorporating sentence-level syntactic information on the target-side, in the NMT decoder. We aim to answer two questions: 1) Is tight integration of target words and syntax better than multitask training? 2) Does target syntax provide complementary information to source syntax for NMT?

We propose a method for tightly coupling words and syntax by interleaving the target syntactic representation, in the form of CCG supertags, with the target word sequence. We compare this to loosely coupling words and syntax using a multitask solution [Luong et al., 2016], where the shared parts of the model are trained to produce either a target sequence of words or supertags. Target syntax is especially beneficial for language pairs where no syntactic resources are available on the source-side, which applies to many low-resource language pairs. For language pairs where syntactic resources are available on both the source and target-side, we show that approaches to incorporate source syntax and target syntax are complementary.

Our results on WMT data show that explicitly modeling target-syntax improves machine translation quality for German→English, a high-resource pair, and for Romanian→English, a low-resource pair. Furthermore, a tight coupling of words and syntax improves translation quality more than multitask training. By combining target-syntax with adding source-side dependency labels in the embedding layer, we obtain a total improvement of 0.9 BLEU for German→English and 1.2 BLEU for Romanian→English.

This chapter is structured as follows. In Section 6.2 we discuss the limitations of NMT and previous work on integrating source or target syntactic information. In Section 6.3 we describe the syntactic representation and different strategies of coupling it with the translated words in the decoder or in the encoder of the NMT system. In Section 6.4 we describe the experimental setup and training parameters for the NMT systems. In Section 6.5 we evaluate the effect of target syntax on overall translation quality and make a finer grained analysis with respect to different linguistic constructions and sentence lengths.

6.2 Background

Sequence-to-sequence neural machine translation (NMT) models [Sutskever et al., 2014, Cho et al., 2014b, Bahdanau et al., 2015] are state-of-the-art on many language-pairs [Sennrich et al., 2016a, Junczys-Dowmunt et al., 2016]. In a detailed analysis, Bentivogli et al. [2016] show that NMT significantly improves over phrase-based SMT, in particular with respect to morphology and word order, but that results can still be improved for longer sentences and syntactic phenomena such as prepositional phrase (PP) attachment. Another study by Shi et al. [2016] shows that the encoder layer of NMT partially learns syntactic information about the source language, however syntactic phenomena such as coordination or PP attachment are poorly modeled.

Syntax has helped in statistical machine translation (SMT) to capture dependencies between distant words that impact morphological agreement, subcategorisation and word order [Galley et al., 2004b, Menezes and Quirk, 2007, Williams and Koehn, 2012, Nădejde et al., 2013, Sennrich, 2015, Nădejde et al., 2016a,b, Chiang, 2007].

There has been some work in NMT on modeling source-side syntax implicitly or explicitly. Kalchbrenner and Blunsom [2013], Cho et al. [2014a] capture the hierarchical aspects of language implicitly by using convolutional neural networks, while Eriguchi et al. [2016] use the parse tree of the source sentence to guide the recurrence and attention model in tree-to-sequence NMT. Luong et al. [2016] co-train a translation model and a source-side syntactic parser which share the encoder. Our multitask models extend their work to attention-based NMT models and to predicting target-side syntax as the secondary task. Sennrich and Haddow [2016] generalize the embedding layer of NMT to include explicit linguistic features such as dependency relations and part-of-speech tags and we use their framework to show source and target syntax provide complementary information.

Applying more tightly coupled linguistic factors on the target for NMT has been previously investigated. Niehues et al. [2016] introduces a factored RNN-based language model for re-scoring an n-best list produced by a phrase-based MT system. In recent work Martínez et al. [2016] propose a factored NMT model which generates lemmas and morphological tags, and then uses these to generate the word form. Unfortunately no real gain was reported for these experiments. In our work, we do not focus on model architectures, and instead we explore the more general problem of including target syntax in NMT: comparing tightly and loosely coupled syntactic information and showing source and target syntax are complementary.

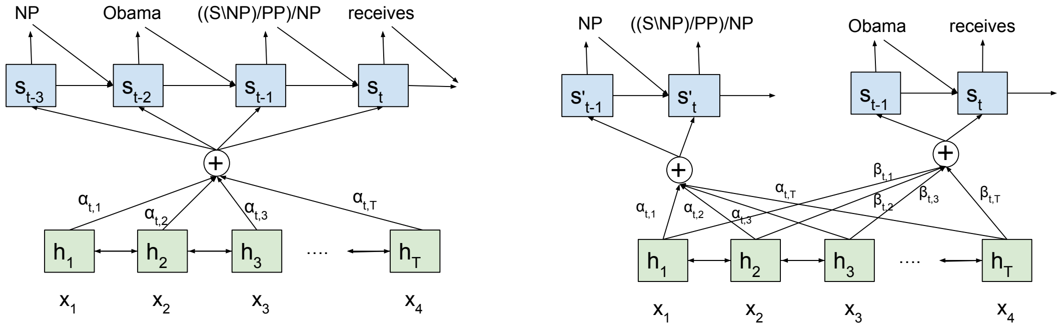
Concurrently with this work, Aharoni and Goldberg [2017] proposed serializing the target constituency trees and Eriguchi et al. [2017] model target dependency relations by augmenting the NMT decoder with a RNN grammar [Dyer et al., 2016]. In our work, we use CCG syntactic categories [Steedman, 2000], also known as *supertags*, to represent syntax explicitly. Supertags provide sentence-level syntactic information locally at the lexical level. They encode subcategorization information, capturing short and long range dependencies and attachments, and also tense and morphological aspects of the word in a given context. Previous work on integrating CCG supertags in factored phrase-based models [Birch et al., 2007] made strong independence assumptions between the target word sequence and the CCG categories. In this work we take advantage of the expressive power of recurrent neural networks to learn representations that generate both words and CCG supertags, conditioned on the entire lexical and syntactic target history.

6.3 Modeling Syntax in NMT

CCG is a lexicalised formalism in which words are assigned syntactic categories, called *supertags*, capturing sentence-level syntactic constraints locally. This word-level syntactic representation is suitable for integration in a sequence-to-sequence NMT systems, either in the encoder or decoder. Although NMT captures long range dependencies using long-term memory, short-term memory is cheap and reliable. Supertags can help by allowing the model to rely more on local information (short-term) and not having to rely heavily on long-term memory. In Section 2.5 we introduced the CCG formalism and gave examples of syntactic information encoded by CCG supertags that may be useful in translation. In this section we propose a method of integrating target-side syntax in the decoder, in the form of CCG supertags, and describe how to combine these with source-side syntax in the encoder.

6.3.1 Baseline decoder

The baseline decoder architecture is a conditional GRU with attention ($cGRU_{attn}$) as implemented in the Nematus toolkit [Sennrich et al., 2017]. The decoder is a recursive function computing a hidden state s_j at each time step $j \in [1,T]$ of the target recurrence. This function takes as input the previous hidden state s_{j-1}, the embedding of the previous target word y_{j-1} and the output of the attention model c_j. The attention model computes a weighted sum over the hidden states $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$ of the bi-directional RNN encoder. The function g computes the intermediate representation t_j and passes this to a *softmax* layer which first applies a linear transformation (W_o) and then computes the probability distribution over the target vocabulary. The training objective for the entire architecture is minimizing the discrete cross-entropy, therefore the loss l is the negative log-probability of the reference sentence.



a) interleaving

b) multitasking

Figure 6.1: Model structure for different methods of integrating target syntax in the neural machine translation (NMT) decoder: a) interleaving and b) multitasking. x_i are the source words, h_i are the hidden states of the encoder and s_j are the hidden states of the decoder.

$$s'_j = GRU_1(y_{j-1}, s_{j-1}) \tag{6.1}$$

$$c_j = ATT([h_1;...;h_{|x|}], s'_j) \tag{6.2}$$

$$s_j = cGRU_{attn}(y_{j-1}, s_{j-1}, c_j) \tag{6.3}$$

$$t_j = g(y_{j-1}, s_j, c_j) \tag{6.4}$$

$$p_y = \prod_{j=1}^{T} p(y_j|x, y_{1:j-1}) = \prod_{j=1}^{T} softmax(t_j W_o) \tag{6.5}$$

$$l = -log(p_y) \tag{6.6}$$

6.3.2 Target-side syntax

When modeling the target-side syntactic information we consider different strategies of coupling the CCG supertags with the translated words in the decoder: interleaving and multitasking with shared encoder. In Figure 6.1 we represent graphically the differences between the two strategies and in the next paragraphs we formalize them.

Interleaving In this paper we propose a tight integration in the decoder of the syntactic representation and the surface forms. Before each word of the target sequence we include its supertag as an extra token. The new target sequence y' will have the length $2T$, where T is the number of target words. With this representation, a single decoder

Source-side

BPE:	Obama	receives	Net+	an+	yahu	in	the	capital	of	USA
IOB:	O	O	B	I	E	O	O	O	O	O
CCG:	NP	((S[dcl]\NP)/PP)/NP	NP	NP	NP	PP/NP	NP/N	N	(NP\NP)/NP	NP

Target-side

NP Obama ((S[dcl]\NP)/PP)/NP receives NP Net+ an+ yahu PP/NP in NP/N the N capital (NP\NP)/NP of NP USA

Figure 6.2: Source and target representation of syntactic information in syntax-aware neural machine translation. For the source side, we show the byte pair encoding (BPE), inside-outside-beginning (IOB) tags, as well as the combinatory categorial grammar (CCG) representation. “Netanyahu” is split into BPE subword units since it does not appear frequently in the training data.

learns to predict both the target supertags and the target words conditioned on previous syntactic and lexical context. We do not make changes to the baseline NMT decoder architecture, keeping equations (6.1) - (6.6) and the corresponding set of parameters unchanged. Instead, we augment the target vocabulary to include both words and CCG supertags. This results in a shared embedding space and the following probability of the target sequence y', where y'_j can be either a word or a tag:

$$y' = y_1^{tag}, y_1^{word}, ..., y_T^{tag}, y_T^{word} \tag{6.7}$$

$$p_{y'} = \prod_{j}^{2T} p(y'_j | x, y'_{1:j-1}) \tag{6.8}$$

At training time we pre-process the target sequence to add the syntactic annotation and then split only the words into *byte-pair-encoding* (BPE) [Sennrich et al., 2016b] sub-units. At testing time we delete the predicted CCG supertags to obtain the final translation. Figure 6.2 gives an example of the target-side representation in the case of interleaving. The supertag *NP* corresponding to the word *Netanyahu* is included only once before the three BPE subunits *Net+ an+ yahu*.

Multitasking – shared encoder A loose coupling of the syntactic representation and the surface forms can be achieved by co-training a translation model with a secondary prediction task, in our case CCG supertagging. In the multitask framework [Luong et al., 2016] the encoder part is shared while the decoder is different for each of the prediction tasks: translation and tagging. In contrast to Luong et al., we train a separate attention model for each task and perform multitask learning with target syntax. The two decoders take as input the same source context, represented by the encoder's hidden states $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$. However, each task has its own set of parameters associated with the five components of the decoder: GRU_1, ATT, $cGRU_{att}$, g, $softmax$. Furthermore, the two decoders may predict a different number of target symbols, resulting in target sequences of different lengths T_1 and T_2. We measure how often this occurs in Section 6.5.4. This results in two probability distributions over separate target vocabularies for the words and the tags:

$$p_y^{word} = \prod_j^{T_1} p(y_j^{word} | x, y_{1:j-1}^{word}) \tag{6.9}$$

$$p_y^{tag} = \prod_k^{T_2} p(y_k^{tag} | x, y_{1:k-1}^{tag}) \tag{6.10}$$

The final loss is the sum of the losses for the two decoders:

$$l = -(log(p_y^{word}) + log(p_y^{tag})) \tag{6.11}$$

We use EasySRL to label the English side of the parallel corpus with CCG supertags[1] instead of using a corpus with gold annotations as in Luong et al. [2016].

6.3.3 Source-side syntax

While our focus is on target-side syntax, we also experiment with including source-side syntax to show that the two approaches are complementary.

Shared embedding Sennrich and Haddow propose a framework for including source-side syntax as extra features in the NMT encoder. They extend the model of Bahdanau et al. by learning a separate embedding for several source-side features such as the word itself or its part-of-speech. All feature embeddings are concatenated into one embedding vector which is used in all parts of the encoder model instead of the word embedding. When modeling the source-side syntactic information, we include the CCG supertags or dependency labels as extra features. The baseline features are the subword units obtained using *byte-pair-encoding* (BPE, [Sennrich et al., 2016b]) together with the annotation of the subword structure using IOB format by marking if a

[1] We use the same data and annotations for the *interleaving* approach.

	train	dev	test
DE-EN	4,468,314	2,986	2,994
RO-EN	605,885	1,984	1,984

Table 6.1: Number of sentences in the training, development and test sets for German (DE)→English (EN) and Romanian (RO)→English language pairs.

symbol in the text forms the beginning (B), inside (I), or end (E) of a word. A separate tag (O) is used if a symbol corresponds to the full word. The word level supertag is replicated for each BPE unit. Figure 6.2 gives an example of the source-side feature representation.

6.4 Experimental Setup

We train the neural MT systems on all the parallel data available at WMT16 [Bojar et al., 2016] for the German↔English and Romanian↔English language pairs. The English side of the training data is annotated with CCG lexical tags[2] using EasySRL [Lewis et al., 2015]. Some longer sentences cannot be processed by the parser and therefore we eliminate them from our training and test data. We report the sentence counts for the filtered data sets in Table 6.1. Dependency labels are annotated with ParZU [Sennrich et al., 2013] for German and SyntaxNet [Andor et al., 2016] for Romanian.

All the neural MT systems are attentional encoder-decoder networks [Bahdanau et al., 2015] as implemented in the Nematus toolkit [Sennrich et al., 2017].[3] We use similar hyper-parameters to those reported by Sennrich et al. [2016a], Sennrich and Haddow [2016] with minor modifications: we used mini-batches of size 60 and Adam optimizer [Kingma and Ba, 2014]. The full list of parameters is given in Appendix B. We select the best single models according to BLEU on the development set and use the four best single models for the ensembles.

To show that we report results over strong baselines, Table 6.2 compares the scores obtained by our baseline system to the ones reported in Sennrich et al. [2016a]. We normalize diacritics for the English→Romanian test set.[4] We did not remove or nor-

[2] The CCG tags include features such as the verb tense (e.g. [ng] for continuous form) or the sentence type (e.g. [pss] for passive).

[3] https://github.com/rsennrich/nematus

[4] There are different encodings for letters with cedilla (ş,ţ) used interchangeably throughout the cor-

malize Romanian diacritics for the other experiments reported in this chapter. Our baseline systems are generally stronger than Sennrich et al. [2016a] due to training with a different optimizer for more iterations.

	This work	Sennrich et al. [2016a]
DE→EN	31.0	28.5
EN→DE	27.8	26.8
RO→EN	28.0	27.8
EN→RO[1]	25.6	23.9

Table 6.2: Comparison of baseline systems in this work and in Sennrich et al. [2016a]. Case-sensitive BLEU scores reported over newstest2016 with *mteval-13a.perl*. [1]Normalized diacritics. This table was adapted from Nădejde et al. [2017].

During training we validate our models with BLEU [Papineni et al., 2002] on development sets: newstest2013 for German↔English and newsdev2016 for Romanian↔English. We evaluate the systems on newstest2016 test sets for both language pairs and use bootstrap resampling [Riezler and Maxwell, 2005] to test statistical significance. We compute BLEU with *multi-bleu.perl* over tokenized sentences both on the development sets, for early stopping, and on the test sets for evaluating our systems.

Words are segmented into sub-units that are learned jointly for source and target using BPE [Sennrich et al., 2016b], resulting in a vocabulary size of 85,000. The vocabulary size for CCG supertags was 500.

For the experiments with source-side features we use the BPE sub-units and the IOB tags as baseline features. We keep the total word embedding size fixed to 500 dimensions. We allocate 10 dimensions for dependency labels when using these as source-side features and when using source-side CCG supertags we allocate 135 dimensions. For the IOB tags we allocate 5 dimensions.

The *interleaving* approach to integrating target syntax increases the length of the target sequence. Therefore, at training time, when adding the CCG supertags in the target sequence we increase the maximum length of sentences from 50 to 100. On average, the length of English sentences for newstest2013 in BPE representation is 22.7, while the average length when adding the CCG supertags is 44.[5] Increasing

pus. `https://en.wikipedia.org/wiki/Romanian_alphabet#ISO_8859`

[5] The CCG supertag is output only once for every target word, and a word can be split in multiple BPE sub-units. This is why the length of the target sequence including CCG supertags is slightly lower than double the length of the BPE sequence

			German→English		Romanian→English	
Model	Syntax	Strategy	Single	Ensemble	Single	Ensemble
NMT	-	-	31.0	32.1	28.1	28.4
SNMT	target – CCG	interleaving	32.0	**32.7***	29.2	**29.3****
Multitasking	target – CCG	shared encoder	31.4	32.0	28.4	29.0*
SNMT	source – dep	shared embedding	31.4	32.2	28.2	28.9
	+ target – CCG	+ interleaving	32.1	**33.0****	29.1	**29.6****

Table 6.3: Results of experiments with target-side syntax for German→English and Romanian→English. BLEU scores reported for baseline NMT, SNMT and the multitasking model. The SNMT system is additionally combined with source dependencies. Statistical significance indicated by * $p < 0.05$ and ** $p < 0.01$. Highest scores in bold. This table was adapted from Nădejde et al. [2017].

the length of the target recurrence results in larger memory consumption and slower training.[6]. At test time, we obtain the final translation by post-processing the predicted target sequence to remove the CCG supertags.

6.5 Evaluation

In this section, we first evaluate the syntax-aware NMT model (SNMT) with target-side CCG supertags as compared to the baseline NMT model described in the previous section [Bahdanau et al., 2015, Sennrich et al., 2016a]. We show that our proposed method for tightly coupling target syntax via *interleaving*, improves translation for both German→English and Romanian→English while the *multitasking* framework does not. Next, we show that SNMT with target-side CCG supertags can be complemented with source-side dependencies, and that combining both types of syntax brings the most improvement. Finally, our experiments with source-side CCG supertags confirm that syntax can improve translation either as extra information in the encoder or in the decoder.

6.5.1 Target-side syntax

We first evaluate the impact of target-side CCG supertags on overall translation quality. In Table 6.3 we report results for German→English, a high-resource language pair, and

[6] Roughly 10h30 per 100,000 sentences (20,000 batches) for SNMT compared to 6h for NMT.

for Romanian→English, a low-resource language pair. We report BLEU scores for both the best single models and ensemble models. However, we will only refer to the results with ensemble models since these are generally better.

The SNMT system with target-side syntax improves BLEU scores by 0.9 for Romanian→English and by 0.6 for German→English. Although the training data for German→English is large, the CCG supertags still improve translation quality. These results suggest that the baseline NMT decoder benefits from modeling the sentence-level syntactic information locally via supertags.

Next, we evaluate whether there is a benefit to tight coupling between the target word sequence and syntax, as opposed to loose coupling. We compare our method of *interleaving* the CCG supertags with *multitasking*, which predicts target CCG supertags as a secondary task. The results in Table 6.3 show that the multitask approach does not improve BLEU scores for German→English, which exhibits long distance word reordering. For Romanian→English, which exhibits more local word reordering, multitasking improves BLEU by 0.6 relative to the baseline. In contrast, the *interleaving* approach improves translation quality for both language pairs and to a larger extent. Therefore, we conclude that a tight integration of the target syntax and word sequence is important. Conditioning the prediction of words on their corresponding CCG supertags is what sets SNMT apart from the multitasking approach. To understand better what is improving when adding target syntax, in Section 6.5.3 we analyze the BLEU score results by sentence length and linguistic constructs found in the sentence. In Section 6.5.4, we evaluate the accuracy of the SNMT system at predicting the target-side CCG supertag sequence.

Contrastive experiments We explore further whether conditioning the prediction of words on their corresponding CCG supertags is essential for improving translation quality. We first vary the way we integrate syntax in the target word sequence by predicting first all CCG supertags and second all the target words. In this case, the syntactic context is encoded in the decoder hidden state, however there is no direct dependency between the target word and its corresponding CCG supertag.

Second, we consider an alternative to multitasking which softens the independence assumptions between target words and CCG supertags: *distinct softmax*. In this approach only the softmax layer is distinct for each task while the encoder, attention model and decoder are shared. The input to the two softmax layers is indentical (the context vector, the previous hidden state and the previous predicted word) but the second softmax layer predicts CCG supertags. The cost of predicting the wrong supertag is added to the cost of predicting the wrong target word.

The BLEU scores for the ensemble models are shown in Table 6.4.

Model	Strategy	German→English	Romanian→English
NMT	-	32.1	28.4
SNMT	interleaving	**32.7**	**29.3**
SNMT	syntax first	31.3	29.3
Multitasking	distinct softmax	32.0	-

Table 6.4: BLEU score results of contrastive experiments with target–syntax which vary the degree of independence between target words and CCG supertags. Highest scores in bold.

The German→English system which predicts the CCG supertag sequence before the target word sequence performs significantly worse than the baseline NMT system. For Romanian→English, this system performs similarly to the system using the interleaving approach. These results suggest that a direct dependency between words and CCG supertags is important for German→English. For Romanian→English, it is sufficient to have the syntactic context encoded in the decoder state.

Finally, the system with distinct softmax layers does not improve translation quality for German→English as compared to the baseline NMT system. This confirms again that a tight coupling of the target word and syntax is needed for this language pair.

6.5.2 Source-side and target-side syntax

We now show that our method for integrating target-side syntax can be combined with the framework of Sennrich and Haddow [2016] for integrating source-side linguistic information, leading to further improvement in translation quality. We evaluate the syntax-aware NMT system, with CCG supertags as target-syntax and dependency labels as source-syntax. While the dependency labels do not encode sentence-level syntactic constraints, they disambiguate the grammatical function of words. Initially, we had intended to use CCG supertags on the source-side as well for German→English, however the German CCG tree-bank is still under development.

From the results in Table 6.3 we first observe that for German→English the source-side dependency labels improve BLEU by only 0.1, while Romanian→English sees an improvement of 0.5. Source-syntax may help more for Romanian→English because the training data is smaller and the word order is more similar between the source and target languages than it is for German→English.

For both language pairs, target-syntax improves translation quality more than source-syntax. However, target-syntax is complemented by source-syntax when used together, leading to a final improvement of 0.9 BLEU points for German→English and 1.2 BLEU points for Romanian→English.

Finally, we show that CCG supertags are also an effective representation of global-syntax when used in the encoder. In Table 6.5 we present results for using CCG supertags as source-syntax in the embedding layer. Because we have CCG annotations only for English, we reverse the translation directions and report BLEU scores for English→German and English→Romanian. The BLEU scores reported are for the ensemble models over newstest2016.

model	syntax	EN→DE	EN→RO
NMT	-	28.3	25.6
SNMT	source – CCG	**29.0****	**26.1****

Table 6.5: BLEU results for English (EN)→German (DE) and English→Romanian (RO) with source-side syntax. The SNMT system uses the CCG supertags of the source words in the embedding layer. *$p < 0.05$. Best results in bold. This table was adapted from Nădejde et al. [2017].

For English→German BLEU increases by 0.7 points and for English→Romanian by 0.5 points. In contrast, Sennrich and Haddow [2016] obtain an improvement of only 0.2 for English→German using dependency labels which encode only the grammatical function of words. These results confirm that incorporating sentence-level syntactic information in the encoder provides complementary information that the baseline NMT model is not able to learn from the source word sequence alone.

6.5.3 Analyses by sentence type

In this section, we make a finer grained analysis of the impact of target-side syntax by looking at a breakdown of BLEU scores with respect to different linguistic constructions and sentence lengths.[7]

We classify English sentences into different linguistic constructions based on the CCG supertags that appear in them. For example, the presence of category (NP\NP)/(S/NP)

[7] Document-level BLEU is computed over each subset of sentences.

indicates a subordinate construction. If multiple linguistic constructs appear in the same sentence, we count it for each of these constructs. However, if a sentence has multiple constructs of the same type, we only count it once. Appendix C shows the regular expression written in Python for grouping sentences by linguistic constructs. Figure 6.3 shows the difference in BLEU points between the syntax-aware NMT system and the baseline NMT system for the following linguistic constructions: coordination *(conj)*, control and raising *(control)*, prepositional phrase attachment *(pp)*, questions and subordinate clauses *(subordinate)*.

These constructs are challenging for machine translation as they involve complex reordering or agreement. As we previously exemplified in Section 2.5, in German, the verb appears at the end of the subordinate clause and its English translation has to be reordered. Prepositional phrases can also be involved in reordering, because the order of verb arguments is more flexible in German. Questions can also involve movement of prepositions as in example a) from Figure 6.5. In this analysis we did not distinguish between questions involving movement and those which do not, and this could be addressed in future work. Finally, the constructs involving coordination or control and raising can involve long-distance agreement.

In the figure we use the symbol "*" to indicate that syntactic information is used on the target (eg. de-en*), or both on the source and target (eg. *de-en*). We report the number of sentences for each category in Table 6.6.

	sub.	qu.	pp	contr.	conj	total
RO↔EN	742	90	1,572	415	845	1,984
DE↔EN	936	114	2,321	546	1,129	2,994

Table 6.6: Frequency of the English sentences with different linguistic constructs for Romanian (RO)→English (EN) and German (DE)→English (EN) test sets. This table was adapted from Nădejde et al. [2017].

With target-syntax, we see consistent improvements across all linguistic constructions for Romanian→English and across all but *control and raising* for German→English. In particular, the increase in BLEU scores for the *prepositional phrase* and *subordinate* constructions suggests that the target word order is improved. In future work, we propose manually evaluating the SNMT systems to confirm that word order is indeed improved.

For German→English, there is a small decrease in BLEU for the *control and raising*

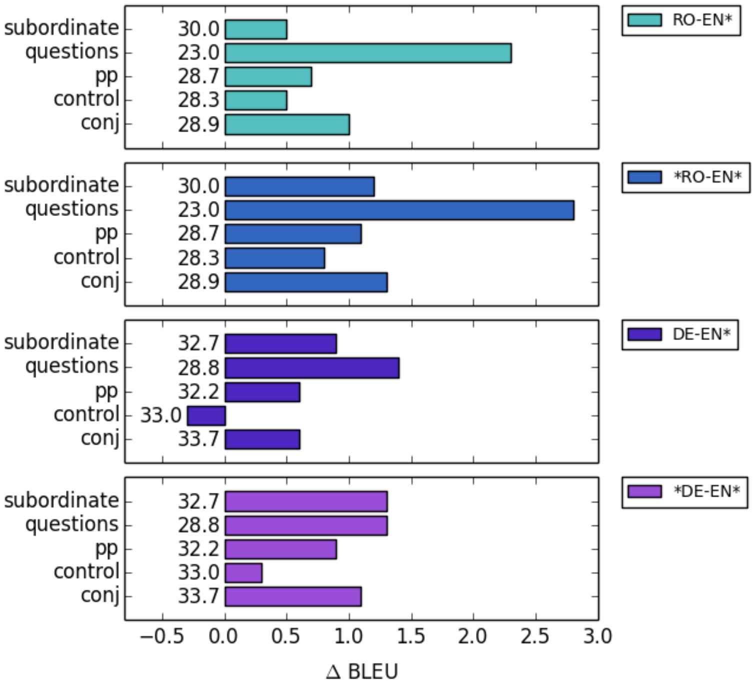


Figure 6.3: Difference in BLEU points between SNMT and NMT (X axis) with regard to different linguistic constructs, relative to baseline NMT scores (shown as labels to the left of the bars). This figure was adapted from Nădejde et al. [2017].

constructions when using target-syntax alone. However, source-syntax adds complementary information to target-syntax, resulting in a small improvement for this category as well. Moreover, combining source and target-syntax increases translation quality across all linguistic constructions as compared to NMT and SNMT with target-syntax alone. For Romanian→English, combining source and target-syntax brings an additional improvement of 0.7 for *subordinate* constructs and 0.4 for *prepositional phrase attachment*. For German→English, on the same categories, there is an additional improvement of 0.4 and 0.3 respectively. Overall, BLEU scores improve by more than 1 BLEU point for most linguistic constructs and for both language pairs.

Next, we compare the systems with respect to sentence length. Figure 6.4 shows the difference in BLEU points between the syntax-aware NMT system and the baseline NMT system with respect to the length of the source sentence measured in BPE sub-units. We report the number of sentences for each category in Table 6.7.

With target-syntax, we see consistent improvements across all sentence lengths for Romanian→English and across all but short sentences for German→English. For

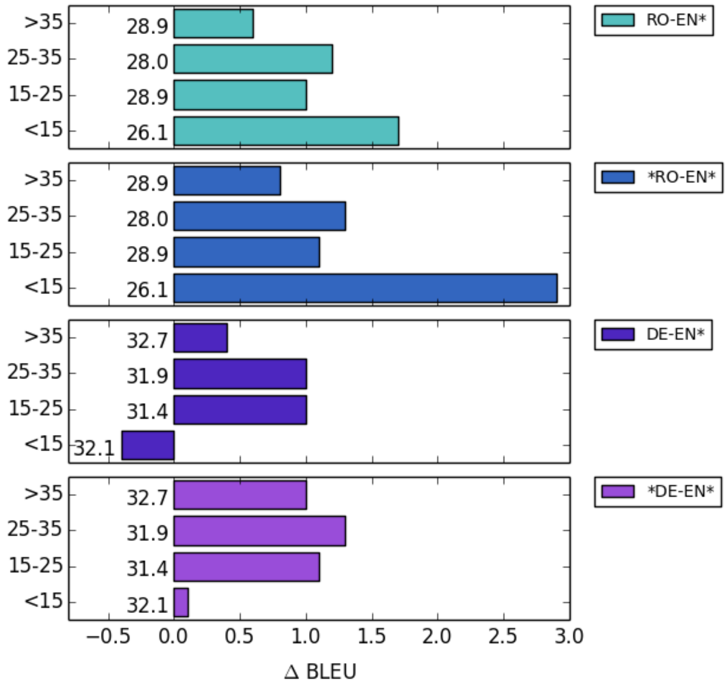


Figure 6.4: Difference in BLEU points (X axis) between SNMT and NMT with regard to sentence length, relative to baseline NMT scores. The BLEU scores for the baseline are shown as label to the left of the bars. This figure was adapted from Nădejde et al. [2017].

German→English there is a decrease in BLEU for sentences up to 15 words. Since the German→English training data is large, the baseline NMT system learns a good model for short sentences with local dependencies and without subordinate or coordinate clauses. Including extra CCG supertags increases the target sequence without adding information about complex linguistic phenomena. However, when using both source and target syntax, the effect on short sentences disappears. For Romanian→English there is also a large improvement on short sentences when combining source and target syntax: 2.9 BLEU points compared to the NMT baseline and 1.2 BLEU points compared to SNMT with target-syntax alone.

With both source and target-syntax, translation quality increases across all sentence lengths as compared to NMT and SNMT with target-syntax alone. For German→English sentences that are more than 35 words, we see again the effect of increasing the target sequence by adding CCG supertags. Target-syntax helps, however BLEU improves by only 0.4, compared to 0.9 for sentences between 15 and 35 words. With both source

	<15	15-25	25-35	>35	total
RO↔EN	491	540	433	520	1,984
DE↔EN	918	934	582	560	2,994

Table 6.7: Frequency of the English sentences with different sentence lengths (in tokens), for Romanian (RO)→English (EN) and German (DE)→English (EN) test sets. This table was adapted from Nădejde et al. [2017].

and target syntax, BLEU improves by 0.8 for sentences with more than 35 words. For Romanian→English we see a similar result for sentences with more than 35 words: target-syntax improves BLEU by 0.6, while combining source and target syntax improves BLEU by 0.8. These results confirm as well that source-syntax adds complementary information to target-syntax and mitigates the problem of increasing the length of the target sequence.

In Table 6.8 we report the breakdown of BLEU scores used for generating the figures.

	German→English			Romanian→English		
	NMT	SNMT	SNMT	NMT	SNMT	SNMT
category		tgt CCG	tgt CCG + src Dep		tgt CCG	tgt CCG + src Dep
conj	33.7	34.3	34.8	28.9	29.9	30.2
control	33.0	32.7	33.3	28.3	28.8	29.1
pp	32.2	32.8	33.1	28.7	29.4	29.8
questions	28.8	30.2	30.1	23.0	25.3	25.8
subordinate	32.7	33.6	34.0	30.0	30.5	31.2
< 15	32.1	31.7	32.2	26.1	27.8	29.0
15-25	31.4	32.4	32.5	28.9	29.9	30.0
25-35	31.9	32.9	33.2	28.0	29.2	29.3
> 35	32.7	33.1	33.7	28.9	29.5	29.7

Table 6.8: Breakdown of BLEU scores with respect to different linguistic constructs and sentence lengths. The scores are reported for the baseline NMT system, the SNMT system with target (abbreviated as tgt) CCG supertags and the SNMT system with both target CCG supertags and source (src) dependency (dep) labels.

a)	**DE - EN Question**
Source	Oder wollen Sie herausfinden , **über** was andere reden ?
Ref.	Or do you want to find out what others are talking **about** ?
NMT	Or would you like to find out **about** what others are talking **about** ?
SNMT	Or do you want to find out what$_{NP/(S[dcl]/NP)}$ others are$_{(S[dcl]\backslash NP)/(S[ng]\backslash NP)}$ talking$_{(S[ng]\backslash NP)/PP}$ **about**$_{PP/NP}$?

b)	**DE - EN Subordinate**
Source	...dass die Polizei jetzt sagt , ..., und dass Lamb in seinem Notruf **Prentiss zwar als seine Frau bezeichnete** ...
Ref.	...that police are now saying ..., and that while Lamb **referred to Prentiss as his wife** in the 911 call ...
NMT	...police are now saying ..., and that in his emergency call **Prentiss he called his wife** ...
SNMT	...police are now saying ..., and that lamb , in his emergency call , **described**$_{((S[dcl]\backslash NP)/PP)/NP}$ **Prentiss as his wife** ....

Figure 6.5: Comparison of baseline Neural Machine Translation (NMT) system and syntax-aware NMT (SNMT) system with target syntax for German→English. Phrases re-ordered correctly in SNMT vs NMT are in bold.

6.5.4 Discussion

Our experiments demonstrate that target-syntax improves translation for two language pairs: German→English and Romanian→English.

In this section, we investigate the impact of CCG supertags on the alignment models and measure the accuracy of the predicted CCG sequences. First, we give two examples of translations in Figure 6.5. In these examples, the syntax-aware NMT system predicting target CCG supertags generates more grammatical translations than the baseline NMT system. We then show the alignment matrices in Figures 6.6 and 6.7.

In the example **DE-EN* Question** the baseline NMT system translates the preposition *"über"* twice as *"about"*. The SNMT predicts the correct CCG supertag for *"what"* which expects to be followed by a sentence and not a preposition: $\mathsf{NP/(S[dcl]/NP)}$. Therefore the SNMT correctly re-orders the preposition *"about"* at the end of the question.

In the example **DE-EN* Subordinate** the baseline NMT system fails to correctly attach *"Prentiss"* as an object and *"his wife"* as a modifier to the verb *"called (bezeichnete)"* in the subordinate clause. In contrast, the SNMT system predicts the correct sub-categorization frame of the verb *"described"* and correctly translates the entire predicate-argument structure.

In Figures 6.6, 6.7 and 6.8 we plot the attention matrices, using heat maps, for the baseline, SNMT with target syntax and multitasking systems. For each target position, the attention weights over the source positions are represented by different shades of blue, with darker shades corresponding to a higher weight. The attention matrices for the SNMT system show that the target word and its corresponding CCG supertag have similar attention weights, focused on the source word of interest.

In Figure 6.6 a), for the baseline system, the attention weight of the second occurrence of the preposition *"about"* is concentrated on the end-of-sentence symbol[8] *"</s>"* and little attention is distributed over the source preposition *"über"*. Therefore, some other part of the decoder is responsible for generating this preposition a second time, which breaks the sentence structure. Figure 6.6 b), shows that the SNMT system predicts the preposition *"about"* only once with a strong attention weight over the corresponding source word.

In Figure 6.7 a), the baseline system does not generate a translation of the subject *"Lamb"* and instead generates the pronoun *"he"* which has a diffused attention over the source words, including the subject *"Lamb"* and the object *"Prentiss"*. In contrast, Figure 6.7 b) shows the SNMT system generating a subject[9] with a strong attention over the corresponding source word. The SNMT system correctly translates the verb *"bezeichnete"* as *"described"* and predicts the corresponding CCG supertag $((S[dcl]\backslash NP)/PP)/NP$ which includes the subject to the left of the verb and direct object and prepositional phrase to the right. In contrast, Figure 6.8 shows that the multitasking system mistranslates the verb as *"called"* and predicts the CCG supertag $(S[dcl]\backslash NP)/PP$ which includes only a subject to the left of the verb and a prepositional phrase to the right. The multitasking system fails to generate a translation for the object *"Prentiss"*.

Next, we compare how accurate the systems are at predicting the CCG supertag sequence using the interleaving and multitasking approaches. If one of the systems learns a more accurate model of target syntax, that can explain the difference in translation quality. Furthermore, if the predicted CCG sequence is correct it can be used in downstream applications, such as multi-lingual question answering.

First, we measure whether the systems are able to predict the correct number of supertags. For German→English, the system using interleaving learns to predict a CCG supertag for every target word for all the sentences in the evaluation set. In contrast, the system using multitasking predicts the correct number of CCG supertags only for 69% of the sentences (2060 out of 2994). For this system, the average absolute difference between the number of predicted words and CCG supertags is 2.5 tokens.

[8] The encoder state corresponding to the end-of-sentence symbol encodes the entire source sentence (processed from left to right). For this reason, most target words will have some attention weight distributed over this position.

[9] The system does not recognize the proper name "Lamm" and mistranslates it as "lamb".

		Interleaving		Multitasking	
Category	CCG	Total count	Accuracy (%)	Total count	Accuracy (%)
Prepositional phrase	PP/NP	4,191	94.9	1,061	67.1
	$((S\backslash NP)\backslash(S\backslash NP))/NP$	1,143	82.0	302	55.0
	$(N\backslash N)/NP$	1,338	79.2	319	55.2
Intransitive verbs	$S\backslash NP$	1,443	90.5	431	65.2
Transitive verbs	$(S\backslash NP)/NP$	2,327	94.5	674	71.2
	$(S\backslash NP)/PP$	1,273	92.5	343	55.4
Ditransitive verbs	$((S\backslash NP)/PP)/NP$	283	89.0	81	53.0
Relative	$(NP\backslash NP)/(S\backslash NP)$	190	97.9	30	53.3
Subordinate	S/S	592	98.1	166	69.3
All supertags		-	95.7	-	73.2

Table 6.9: Accuracy for CCG supertags representing prepositional phrase attachment, the subcategorization frame of verbs (intransitive, transitive, ditransitive), and subordinate constructs. We only consider sentences for which the system produced the same number of words and CCG supertags and report the total count for each category.

For Romanian→English, the system using interleaving predicts the wrong number of supertags for 6% of the sentences (115 out of 1984), with an average difference of 1.5 tokens. In contrast, the system using multitasking predicts the wrong number of supertags for 81% of the sentences (1613 out of 1984), with an average difference of 4 tokens.

Next, we evaluate the accuracy of the CCG sequences predicted by the two systems, at token level and only for the sentences which have the same number of predicted supertags and words. As "gold" annotations we use the CCG sequence obtained by parsing the predicted word sequences with EasySRL. For the interleaving approach, we obtain the following accuracies: 95.7 for German→English and 96.0 for Romanian→English. For the multitasking approach, the accuracies are considerably lower: 73.2 for German→English and 73.7 for Romanian→English.

In Table 6.9, we report the breakdown of CCG accuracy for German→English over supertags representing prepositional phrase attachment, the subcategorization frame of verbs, and subordinate and relative clauses[10]. We observe that the system using interleaving has an accuracy higher than 90% for most categories and lower accuracies of 79% for $(N\backslash N)/NP$[11] (prepositional phrase attaching to noun phrase) and of 89%

[10] When we count the CCG supertags we ignore features such as *dcl, pss*.

[11] We aggregate the numbers for $(N\backslash N)/NP$ and $(NP\backslash NP)/NP$.

for $((S\backslash NP)/PP)/NP$ (ditransitive verb). For the multitasking approach, the variance across categories is much higher and half of the categories have accuracies below 60%. However, this evaluation potentially penalizes the multi-tasking system since the two decoders are not synchronized. It is possible that the 1-best CCG supertag sequence predicted by the first decoder is correct, but it corresponds to a translation other than the 1-best word sequence predicted by the second decoder[12].

The analyses presented here show that the SNMT system using interleaving is able to predict with high accuracy the CCG supertags disambiguating prepositional phrase attachment, the subcategorization frames of verbs and identifying subordinate sentences. These aspects of syntax contribute to generating better target word order and well-formed predicate-argument structures and this could be confirmed by manual analysis in future work. Future work could also focus on improving translation for categories for which the CCG supertagging accuracy is lower, such as ditransitive verbs and prepositional phrases attaching to nouns.

6.6 Conclusions

In this chapter, we introduced a method to incorporate explicit target-syntax in a neural machine translation system, by interleaving target words with their corresponding CCG supertags. Earlier work on syntax-aware NMT mainly modeled syntax in the encoder, while our experiments suggest modeling syntax in the decoder is also useful. The results we presented in Section 6.5 show that a tight integration of syntax in the decoder improves translation quality for both German→English and Romanian→English language pairs, more so than a loose coupling of target words and syntax as in multitask learning. Finally, by combining our method for integrating target-syntax with the framework of Sennrich and Haddow [2016] for source-syntax we obtain the most improvement over the baseline NMT system: 0.9 BLEU for German→English and 1.2 BLEU for Romanian→English. In particular, the results in Section 6.5.3 show large improvements for longer sentences involving syntactic phenomena such as subordinate and coordinate clauses. By representing target syntax with CCG supertags, which encode subcategorization information, capturing long distance dependencies and attachments, we also improve translation of prepositional phrases, the most frequent type of predicate arguments.

[12] In such cases, a different gold annotation of the CCG supertag sequence should be considered, however it would be impossible to guess wich n-best translation to annotate.

Incorporating global source or target syntax in SMT has been extensively explored, as we have done in Chapters 4 and 5, but there is ongoing work on this topic for NMT. In this chapter we have shown that combining sentence-level lexical and syntactic information from both the source-side and the target-side also improves NMT in particular when long distance dependencies are involved. In future work we propose to manually evaluate sentence pairs which involve reordering, such as those with subordinate clauses, to confirm that the syntactic information improves word order. Future work could also test whether the words receive CCG supertags that are appropriate given the context and can be combined into a well-formed S-rooted tree structure. This analysis could also investigate if a grammatically well-formed CCG supertag sequence discourages repetitions or omissions of words. Another research direction could be to evaluate the impact of target-syntax when translating into a morphologically rich language, for example by using the Hindi CCGBank [Ambati et al., 2016]. Our results with source and target syntax could be improved by exploring other source-side linguistic features. In the next chapter, we summarize the contributions of this thesis and present some future research directions.

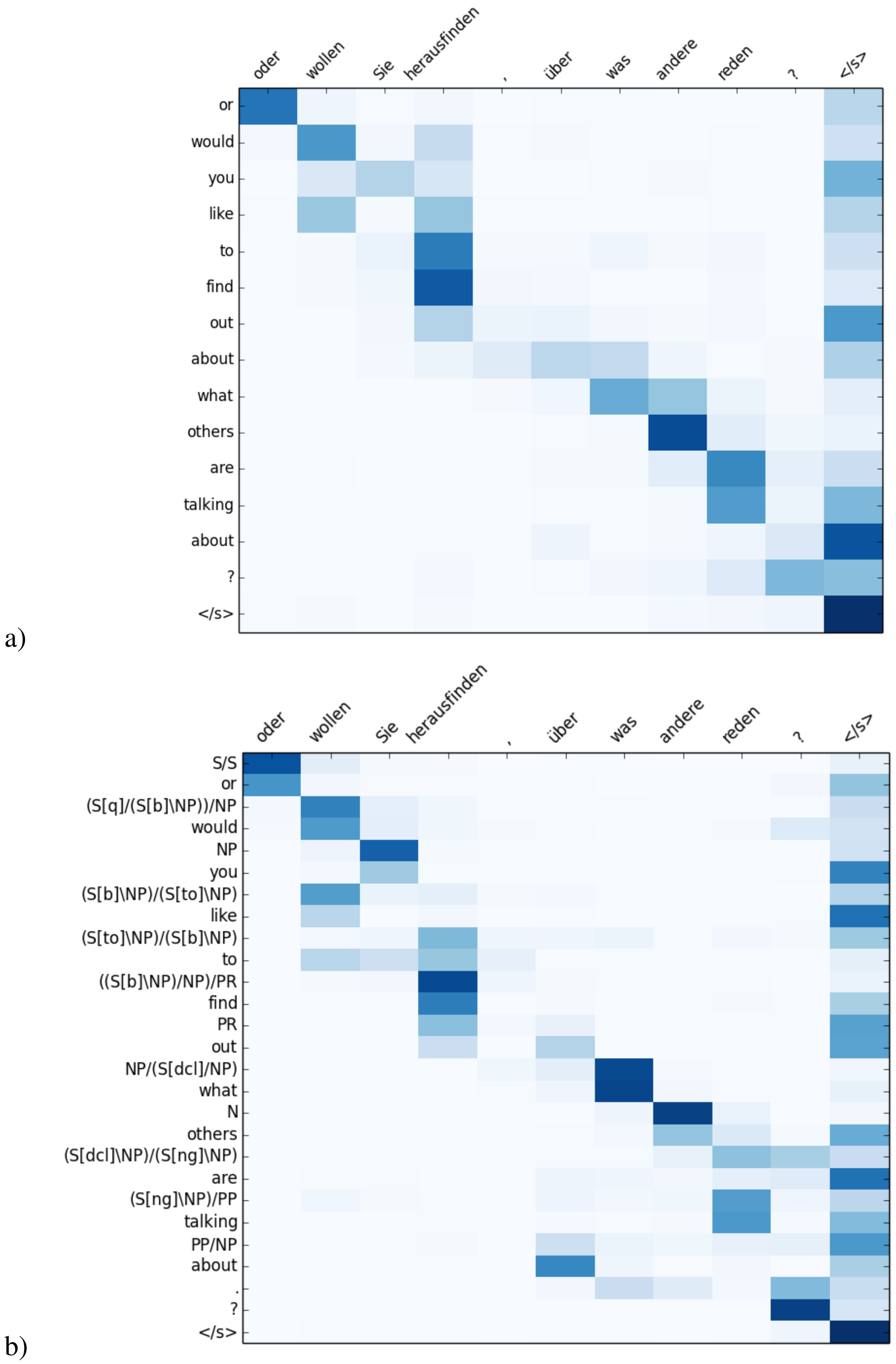


Figure 6.6: Comparison of alignment matrices for a) the baseline NMT system and b) the SNMT system with target syntax. The alignment matrices correspond to the example a) in Figure 6.5. The source sentence (English) is on the Y axis, the target (German) on the X axis.

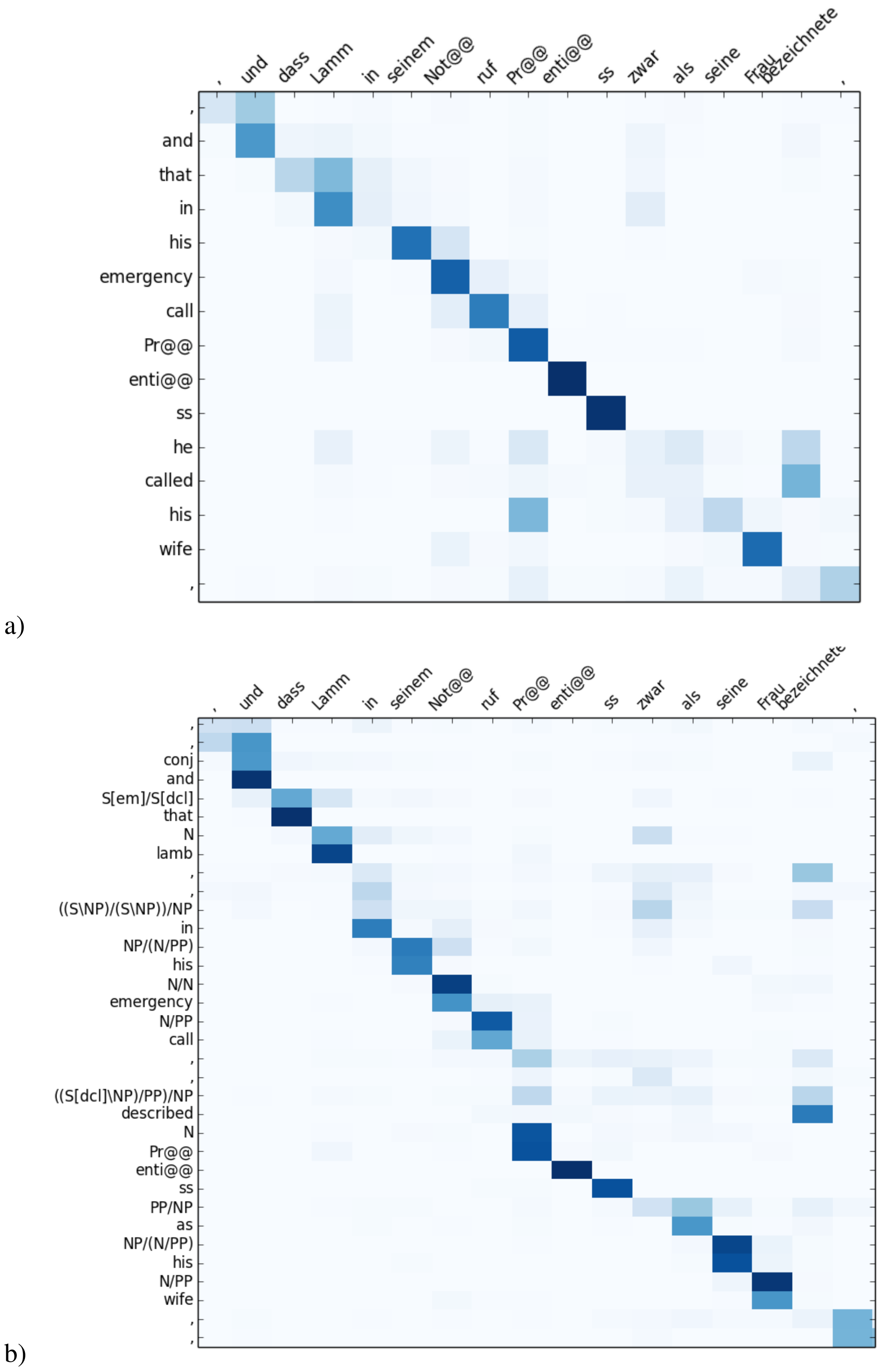


Figure 6.7: Comparison of alignment matrices for a) the baseline NMT system and b) the SNMT system with target syntax. The alignment matrices are cropped and correspond to the subordinate clause in the example b) in Figure 6.5.

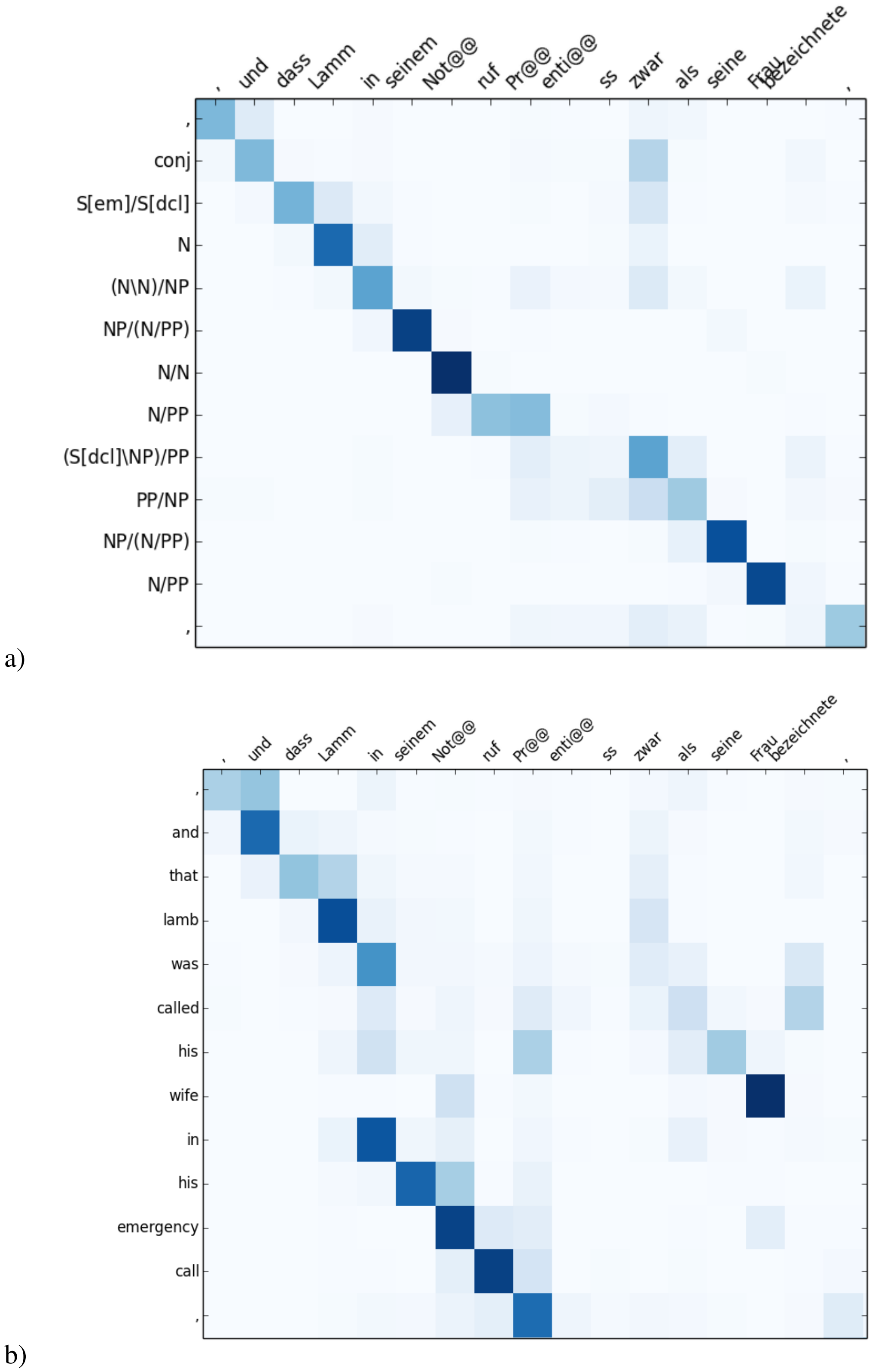


Figure 6.8: The alignment matrices are cropped and correspond to the subordinate clause in the example b) in Figure 6.5.

Chapter 7

Conclusions and Future Work

7.1 Summary

In this thesis we explored how the syntactic structures of the source and target sentences can be leveraged to improve translation of complex syntactic phenomena involving long distance dependencies.

String-to-tree SMT systems use explicit target syntax to handle long-distance reordering, but make strong independence assumptions which lead to inconsistent lexical choices. To address this, we proposed modeling target-side selectional preferences of predicates for their argument fillers, but this was not effective in string-to-tree systems. However, incorporating the global source-side syntactic context in a neural network lexicon model was essential to improving verb translation.

In contrast to SMT, neural machine translation does not make strong independence assumptions thus generating more fluent translations and capturing some long-distance dependencies. Despite the strong learning capabilities of NMT, incorporating additional linguistic information can still improve translation quality. We examined the benefit of incorporating explicit syntactic information on the target-side, showing that tightly coupling words and syntax is most effective at improving translation both in high-resource and low-resource scenarios. Furthermore, we showed that combining target-side and source-side syntactic information brings additive improvements which are consistent across difficult linguistic constructs and sentence lengths.

7.2 Conclusions

We began our work with improving robustness of string-to-tree SMT systems, which were shown to be effective for language pairs exhibiting long-distance word reordering. One important contribution of this thesis was to show this system can achieve state-of-the-art results for German→English on large scale evaluation campaigns [Nădejde et al., 2013, Williams et al., 2014]. Still, our error analyses indicated problems with translating the semantic frames of verbs, which are caused by strong independence assumptions.

We proposed leveraging the target syntactic context available in the decoder to model the semantic affinities between target verbs and their argument fillers. We used dependency relations to represent the predicate argument structure and the selectional association measure proposed by Resnik [1996] to quantify the degree of semantic affinity. Based on these, we introduced a Selectional Preferences feature in a dependency-based string-to-tree system [Nădejde et al., 2016a]. We evaluated three variants of our features, as well a variant of the neural relational dependency language model (RDLM) [Sennrich, 2015] on German→English and did not find significant improvements in automatic metrics. These results prompted an analysis of the *(predicate, argument, dependency relation)* triples, which are scored by our feature. We found that our feature is not effective when the predicate and its arguments are close to each other. In case of the predicates and arguments which are further apart, the translation quality decreases drastically. Furthermore, verbs are often mistranslated which means the conditioning context of our feature is wrong most of the time.

We then performed an in depth analysis of verb translation for a German→English string-to-tree SMT system, that showed grave deficiencies: verb translation recall is as low as 45% and 20% of the main verbs are translated without lexical context. To improve verb translation, we proposed a Neural Verb Lexicon Model trained with a feed-forward neural network and incorporating syntactic context from the source sentence [Nădejde et al., 2016b]. The syntactic context includes all the core arguments of the source verb which carry most semantic information relevant to verb disambiguation. This intuition is confirmed by the improvement in model accuracy of 1.5% over a baseline incorporating only a window context centered on the source verb. When used as an extra feature for re-ranking the output of a string-to-tree system, the NVLM improves verb translation precision by up to 2.7% and recall by up to 7.4%. The syntactic context helped improve precision as compared to the window context, but improved recall to the same extent. Furthermore, these improvements came at the cost of a small (less than 0.5%) decrease in BLEU score.

While the NVLM improves some aspects of translation, other syntactic and lexical inconsistencies appear which are not being addressed with a linear combination of independent models. In contrast, neural machine translation (NMT) does not make such independence assumptions since it incorporates the entire source sentence and target history as context when predicting the next target word. Even though NMT models are able to partially learn source-side syntactic information from sequential lexical information, explicit linguistic features can still improve translation quality. While others proposed incorporating additional source-side linguistic information in NMT, our work was the first to explore the benefit of incorporating target syntax.

We proposed a novel method to incorporate explicit target-syntax in a neural machine translation system, by interleaving target words with their corresponding CCG supertags [Nădejde et al., 2017]. We chose this representation because CCG supertags provide sentence-level syntactic information locally at the lexical level. We then showed that a tight integration of target syntax in the NMT decoder improves translation quality for both German→English and Romanian→English language pairs, more than a loose coupling of target words and syntax as in multitask learning. When incorporating both source and target syntax we obtained additive improvements for both language pairs over a strong baseline NMT system: 0.9 BLEU for German→English and 1.2 BLEU for Romanian→English. Finally, we presented a fine grained analyses showing consistent improvements across difficult linguistic constructs and sentence lengths.

7.3 Contributions

The contributions of this thesis are:

- We explored different methods for improving robustness of string-to-tree systems and build a state-of-the art system for German→English.
- We proposed a Selectional Preferences Model which captures semantic affinities between target predicates and their arguments. We showed that the performance of the model as a feature in a string-to-tree systems for German→English suffers because of overlap with the language model and because of mistranslated verbs.
- We presented an analysis of verb translation in string-to-tree systems for

German→English highlighting that verb translation recall is as low as 45% and that 20% of the main verbs are translated without lexical context.

- We proposed a Neural Verb Lexicon Model to address the problem of mistranslated verbs in string-to-tree systems. The model uses a rich source-side syntactic context, including the subcategorization frame, improving verb translation precision by up to 2.7% and recall by up to 7.4%.

- We proposed a novel method to incorporate explicit target-syntax in a neural machine translation system, by interleaving target words with their corresponding CCG supertags. We showed that target language syntax improves translation quality in both high-resource and low-resource scenarios, and that a tight coupling of target words and syntax (by interleaving) is better than a loose coupling as in multitask learning.

- We showed that by combining our method for Syntax-aware NMT (SNMT) with target CCG supertags with a framework for incorporating source-side linguistic information, we obtain the most improvement in translation quality.

- We presented a fine grained analysis of SNMT and show consistent gains when looking at different linguistic phenomena and sentence lengths.

7.4 Future Work

In Chapter 4.1 we argued that the effectiveness of the selectional preference feature may be limited by: errors in the target syntactic trees generated by the system and mistranslated verbs. We propose in future work to manually identify system translations exhibiting several attachment errors and to use these sentences to re-evaluate the selectional preference feature. To measure the impact of mistranslated verbs, we propose as future work an oracle experiment in which we force the system with the selectional preference feature to generate the correct verb (if the correct translation can be reached).

In Chapter 5.1 we observed that, in some cases, the verb translation is improved by the NVLM but at the cost of other errors appearing in the translation. Future work could address this issue by integrating the NVLM as a feature in the string-to-tree decoder. Another line of research could investigate whether the NVLM has more impact on translation quality if it also predicts translations for predicative nouns. Precision could be improved further if additional target-side context is provided to the NVLM, for example by integrating it with the selectional preference feature in the decoder. Tuning feature weights towards a metric which combines recall and precision, such as METEOR , or towards a metric that considers syntactic n-grams, such as HWCM , could results in more gains in verb translation recall.

This thesis also paves the way for future work on incorporating target-side linguistic information in NMT and we suggest next a few possible directions. A first step should be to manually evaluate the SNMT systems with target-syntax on sentences involving reordering, to confirm whether word order is indeed improved and for which linguistic constructs (e.g. subordinate clauses).

A natural extension to the current set of experiments would be evaluating the impact of target syntax for translation into morphologically rich languages. We suggested at the end of Chapter 6 to experiment with English→Hindi, using the Hindi CCG-Bank [Ambati et al., 2016]. Still, a more large scale survey of target syntax in NMT could be performed using dependency labels instead of CCG supertags, as dependency parsers are available for several languages [Andor et al., 2016]. Although dependency labels do not encode sentence-level syntactic information at the lexical level, they do disambiguate the syntactic function of words. Other aspects of translation could also be evaluated, for example agreement between subject and verb in the presence of multiple attractors similar to the monolingual analysis performed for LSTMs by Linzen et al. [2016].

One limitation of the current work is that interleaving the syntactic representation with the target words results in doubling the length of the target recurrence. A more efficient decoder architecture could deal with this shortcoming, for example by having two distinct softmax layers with a dependency between the different linguistic factors. Martínez et al. [2016] explored a few architectures for a factored decoder but only evaluated these in the context of predicting two factors, the lemma and a morphological tag, to reduce the size of the target vocabulary. While they did not report any significant improvements, there is hope that including CCG factors and BPE sub-units would achieve similar improvements as reported in this thesis, but with a lower computational cost.

A factored architecture would also allow adding other linguistic information such as the part-of-speech or dependency labels which were successfully used as source-side factors [Sennrich and Haddow, 2016]. This would be especially beneficial for language pairs where no syntactic resources are available on the source-side, which applies to many low-resource language pairs. The difficulty with this approach would be to design a strategy for synchronizing the word level linguistic annotation with the BPE sub-units. When using linguistic annotation in the embedding layer, Sennrich and Haddow [2016] duplicate the factors for each BPE sub-unit. However, this might not be the best approach in the decoder if the BPE factor is conditioned on the other linguistic factors at each time step, since this would require a cross-product between all the factors.

Appendix A

Training Data Statistics

Below we report the number of sentences in the parallel training data and test sets for the WMT 2013-2016 shared tasks [Federmann et al., 2013, Bojar et al., 2014, 2015, 2016].

Year	Europarl	News Commentary	Common Crawl	Test set
2013	1,920,209	178,221	2,399,123	3,000
2014	1,920,209	201,288	2,399,123	3,003
2015	1,920,209	216,190	2,399,123	2,169
2016	1,920,209	242,770	2,399,123	2,999

Appendix B

Nematus Parameters

Below we list the parameters used for training the NMT systems with the Nematus toolkit.

- Baseline NMT system

Parameter name	DE→EN	RO→EN
dim_word	500	500
factors	1	1
dim_per_factor	[500]	[500]
dim	1024	1024
n_words	85000	85000
n_words_src	85000	85000
decay_c	0.	0.1
clip_c	1.	1.
lrate	0.0001	0.0001
optimizer	'adam'	'adam'
maxlen	50	50
batch_size	50	60
valid_batch_size	50	60
use_dropout	False	True
dropout_embedding	-	0.2
dropout_hidden	-	0.2
dropout_source	-	0.1
dropout_target	-	0.1

- For the NMT systems with target CCG supertags the only parameter change is:

  $maxlen = 100$

- For NMT systems with source side factors, [word, IOB tag, dependency label] the following parameters change:

  $factors = 3$

  $dim\_per\_factor = [485, 5, 10]$

- For NMT systems with source side factors, [word, IOB tag, CCG supertag] the following parameters change:

  $factors = 3$

  $dim\_per\_factor = [360, 5, 135]$

Appendix C

Selecting sentences by type

Python code for grouping English sentences according to linguistic constructs identified with the CCG supertags:

```
for i, line in enumerate(sys.stdin):
    if line.find("PP") != -1 or line.find("((S\NP)/(S\NP))/NP") != -1 \
or line.find("(NP\NP)/NP") != -1:
        pp.append(i)
    if line.find("S[q]") != -1 or line.find("S[wq]") != -1 \
or line.find("S[qem]") != -1:
        questions.append(i)
    if line.find("conj") != -1:
        conj.append(i)
    if re.search(\
"\(N[P]?[\\\/]N[P]?\)[\\\/]\(S(\[[^\(\)\\\/]+\])?[\\\/]N[P]?\)", line) \
or re.search("\|S(\[[^\(\)\\\/]+\])?[\\\/]S(\[[^\(\)\\\/]+\])? ", line):
        relatives_and_subordinates.append(i)
    if re.search(\
"\(S(\[[^\(\)\\\/]+\])?[\\\/]N[P]?\)\/\(S(\[to\])[\\\/]N[P]?\)", line):
        control.append(i)
```

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)*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

A. V. Aho and J. D. Ullman. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3(1):37–56, February 1969. ISSN 0022-0000.

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. Hindi CCGbank: CCG Treebank from the Hindi Dependency Treebank. In *Language Resources and Evaluation*, 2016.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1231`.

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

Paul Baltescu, Phil Blunsom, and Hieu Hoang. Oxlm: A neural language modelling framework for machine translation. *The Prague Bulletin of Mathematical Linguistics*, 102(1):81–92, october 2014. URL `https://ufal.mff.cuni.cz/pbml/102/art-baltescu-blunsom-hoang.pdf`.

Marzieh Bazrafshan and Daniel Gildea. Semantic roles for string to tree machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL 2013, pages 419–423, Sofia, Bulgaria, 2013.

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`.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 257–267, 2016.

Alexandra Birch. *Reordering Metrics for Statistical Machine Translation*. PhD thesis, School of Informatics, University of Edinburgh, 2011.

Alexandra Birch, Miles Osborne, and Philipp Koehn. Ccg supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 9–16, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1626355.1626357`.

Alexandra Birch, Barry Haddow, Ulrich Germann, Maria Nădejde, Christian Buck, and Philipp Koehn. The feasibility of HMEANT as a human MT evaluation metric. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 52–61, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W13-2203`.

Alexandra Birch, Matthias Huck, Nadir Durrani, Nikolay Bogoychev, and Philipp Koehn. Edinburgh slt and mt system description for the iwslt 2014 evaluation. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, USA, 2014.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical*

Machine Translation, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W14/W14-3302.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September 2015.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics.

Fabienne Braune, Nina Seemann, and Alexander Fraser. Rule selection with soft syntactic features for string-to-tree statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1095–1101, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL https://aclweb.org/anthology/D/D15/D15-1129.

Fabienne Braune, Alexander Fraser, Hal Daumé III, and Aleš Tamchyna. A framework for discriminative rule selection in hierarchical moses. In *Proceedings of the First Conference on Machine Translation*, pages 92–101, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-2210.

Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 61–72, 2007.

Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods* *in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October 2014a. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, 2014b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1082`.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University, 1998.

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*, NAACL HLT '12, pages 427–436, Montreal, Canada, 2012.

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, 2005.

David Chiang. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, 2007.

Rohan Chitnis and John DeNero. Variable-length word encodings for neural translation models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2088–2093, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D/D15/D15-1249`.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W14-4012`.

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the*

2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar, October 2014b. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1160`.

Raphael Cohen, Yoav Goldberg, and Michael Elhadad. Domain adaptation of a dependency parser with a class-class selectional preference model. In *Proceedings of ACL 2012 Student Research Workshop*, ACL '12, pages 43–48, 2012.

Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 531–540, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219906. URL `https://doi.org/10.3115/1219840.1219906`.

Tim Van De Cruys. A neural network approach to selectional preference acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL '14, pages 26–35, 2014.

Marie-Catherine de Marneffe and Christopher D. Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 1–8. Association for Computational Linguistics, 2008.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. What Can Syntax-based MT Learn from Phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 755–763, 2007.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 1370–1380, Baltimore, MD, USA, June 2014.

Nadir Durrani, Barry Haddow, Kenneth Heafield, and Philipp Koehn. Edinburgh's machine translation systems for European language pairs. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 114–121, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W13-2212`.

Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh?s phrase-based machine translation systems for wmt-14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 97–104, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W14/W14-3309`.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California, June 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N16-1024`.

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`.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Katrin Erk, Sebastian Padó, and Ulrike Padó. A flexible, corpus-driven model of regular and inverse selectional preferences. *Comput. Linguist.*, 36(4):723–763, 2010.

Manaal Faruqui and Sebastian Padó. Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany, 2010.

Christian Federmann, Christian Buck, Barry Haddow, Chris Callison-burch, Lucia Specia, and Matt Post. Findings of the 2013 Workshop on Statistical Machine Translation. pages 1–44, 2013.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, 2005.

Heidi J Fox. Phrasal Cohesion and Statistical Machine Translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 304–311, 2002.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *HLTNAACL 2004 Main Proceedings*, pages 273–280, 2004a.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '04, 2004b.

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 the 44th annual meeting of the ACL ACL 06*, pages 961–968, 2006a. doi: 10.3115/1220175.1220296.

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 *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968, Morristown, NJ, USA, 2006b.

Qin Gao and Stephan Vogel. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. Utilizing target-side semantic role labels to assist hierarchical phrase-based machine translation. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 107–115, Portland, Oregon, USA, 2011.

A. Gastel, S. Schulze, Y. Versley, and E. Hinrichs. Annotation of explicit and implicit discourse relations in the t?ba-d/z treebank. In *Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*, Hamburg, Germany, 2011.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. Lexical translation model using a deep neural network architecture. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT 2014)*, 2014.

Barry Haddow, Matthias Huck, Alexandra Birch, Nikolay Bogoychev, and Philipp Koehn. The edinburgh/jhu phrase-based machine translation systems for wmt 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 126–133, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/W15-3013`.

Teresa Herrmann, Jan Niehues, and Alex Waibel. Source discriminative word lexicon for translation disambiguation. In *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT 2015)*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. Montreal neural machine translation systems for wmt?15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/W15-3014`.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. In *Proceedings of the IWSLT 2016*, December 2016.

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D13-1176`.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Eliyahu Kiperwasser and Yoav Goldberg. Semi-supervised dependency parsing using bilexical contextual features from auto-parsed data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1348–1353, September 2015.

Kevin Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25:607–615, 1999. ISSN 08912017.

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*, 2007.

Jonathan K Kummerfeld, David Hall, and James R Curran. Parser Showdown at the Wall Street Corral : An Empirical Investigation of Error Types in Parser Output. In *EMNLP*, number July, pages 1048–1059, 2012a.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1048–1059, 2012b.

Alon Lavie and Michael Denkowski. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 2009.

Mike Lewis, Luheng He, and Luke Zettlemoyer. Joint a* ccg parsing and semantic role labelling. In *Empirical Methods in Natural Language Processing*, 2015.

Junhui Li, Philip Resnik, and Hal Daume. Modeling syntactic and semantic structures in hierarchical phrase-based translation. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics,*, number June in NAACL-HLT 2013, pages 540–549, Atlanta, Georgia, USA, 2013.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

Ding Liu and Daniel Gildea. Syntactic features for evaluation of machine translation. In *ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, MI, 2005.

Ding Liu and Daniel Gildea. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 716–724, 2010.

Qun Liu, Zhongjun He, Yang Liu, and Shouxun Lin. Maximum entropy based rule selection model for syntax-based statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’08, pages 89–97, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1613715.1613729`.

Chi-kiu Lo and Dekai Wu. MEANT : An inexpensive , high-accuracy , semi-automatic metric for evaluating translation utility via semantic frames. In *Proceedings of ACL*, pages 220–229, 2011.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. In *Proceedings of International Conference on Learning Representations (ICLR 2016)*, 2016.

Mercedes García Martínez, Loïc Barrault, and Fethi Bougares. Factored Neural Machine Translation Architectures. In *International Workshop on Spoken Language Translation (IWSLT’16)*, Seattle, USA, 2016.

Arne Mauser, Saša Hasan, and Hermann Ney. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proceedings of the 2009*

Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1, EMNLP '09, pages 210–218, 2009. ISBN 978-1-932432-59-6.

Arul Menezes and Chris Quirk. Using dependency order templates to improve generality in translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 1–8, 2007.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, September 2010.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 95–100, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Jan Niehues and Alex Waibel. An mt error-driven discriminative word lexicon using sentence structure features. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 512 – 520, 2013.

Jan Niehues, Thanh-Le Ha, Eunah Cho, and Alex Waibel. Using factored word representation in neural network language models. In *Proceedings of the First Conference on Machine Translation*, Berlin, Germany, 2016.

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, 2005. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, 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.

Maria Nădejde, Philip Williams, and Philipp Koehn. Edinburgh's Syntax-Based Machine Translation Systems. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 170–176, Sofia, Bulgaria, August 2013.

Maria Nădejde, Alexandra Birch, and Philipp Koehn. Modeling selectional preferences of verbs and nouns in string-to-tree machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 32–42, Berlin, Germany, August 2016a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W16/W16-2204`.

Maria Nădejde, Alexandra Birch, and Philipp Koehn. A neural verb lexicon model with source-side syntactic context for string-to-tree machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, December 2016b.

Maria Nădejde, Reddy Siva, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. Predicting target language ccg supertags improves neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

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`.

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*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, 2014.

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 the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Philip Resnik. Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, 61:127–159, 1996.

Stefan Riezler and John T. Maxwell. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W05-0908`.

Alan Ritter, Mausam, and Oren Etzioni. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 424–434, Stroudsburg, PA, USA, 2010.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994.

Sabine Schulte im Walde. Experiments on the automatic induction of german semantic verb classes. *Comput. Linguist.*, 32(2):159–194, June 2006.

Holger Schwenk. Continuous space translation models for phrase-based statistical machine translation. In *COLING*, 2012.

Holger Schwenk, Daniel Dchelotte, and Jean-Luc Gauvain. Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 723–730, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1273073.1273166`.

Diarmuid Ó. Séaghdha. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 435–444, Stroudsburg, PA, USA, 2010.

Rico Sennrich. Modelling and Optimizing on Syntactic N-Grams for Statistical Machine Translation. *Transactions of the Association for Computational Linguistics*, 3:169–182, 2015.

Rico Sennrich. How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain, 2017. URL `http://aclweb.org/anthology/E17-2060.pdf`.

Rico Sennrich and Barry Haddow. A Joint Dependency Model of Morphological and Syntactic Structure for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, 2015. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany, August 2016.

Rico Sennrich, Martin Volk, and Gerold Schneider. Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2013*, pages 601–609, Hissar, Bulgaria, 2013.

Rico Sennrich, Philip Williams, and Matthias Huck. A tree does not make a well-formed sentence: Improving syntactic string-to-tree statistical machine translation with more linguistic knowledge. *Computer Speech & Language*, 32(1):27–45, 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany, August 2016a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W16/W16-2323`.

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)*, Berlin, Germany, August 2016b. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/E17-3017`.

Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics.

Ekaterina Shutova, Niket Tandon, and Gerard de Melo. Perceptually grounded selectional preferences. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL '15, pages 950–960, 2015.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. TER-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 2009.

Mark Steedman. *The syntactic process*, volume 24. MIT Press, 2000.

Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *Intl. Conf. Spoken Language Processing, Denver, Colorado.*, 2002.

Sara Stymne. Blast: A tool for error analysis of machine translation output. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, HLT '11, pages 56–61, Portland, Oregon, 2011.

Lin Sun and Anna Korhonen. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of the 2009 Conference on Empirical Meth-*

ods in Natural Language Processing: Volume 2 - Volume 2, EMNLP '09, pages 638–647, 2009.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, 2014.

Ales Tamchyna, Alexander M. Fraser, Ondřej Bojar, and Marcin Junczys-Dowmunt. Target-side context for discriminative models in statistical machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1704 – 1714, 2016.

Haiqing Tang, Deyi Xiong, Min Zhang, and Zhengxian Gong. Improving statistical machine translation with selectional preferences. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2154–2163, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `http://aclweb.org/anthology/C16-1203`.

Gertjan van Noord. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the 10th International Conference on Parsing Technologies*, IWPT '07, pages 1–10, 2007.

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, WA, USA, 2013.

Wei Wang, Kevin Knight, Daniel Marcu, and Marina Rey. Binarizing Syntax Trees to Improve Syntax-Based Machine Translation Accuracy. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 746–754, 2007.

Marion Weller, Alexander Fraser, and Sabine Schulte im Walde. Using subcategorization knowledge to improve case prediction for translation to german. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–603, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Marion Weller, Sabine Schulte Im Walde, and Alexander Fraser. Using noun class information to model selectional preferences for translating prepositions in smt. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, AMTA '14, pages 275–287, Vancouver, BC, 2014.

Philip Williams and Philipp Koehn. Agreement constraints for statistical machine translation into german. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 217–226, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1. URL `http://dl.acm.org/citation.cfm?id=2132960.2132990`.

Philip Williams and Philipp Koehn. Ghkm rule extraction and scope-3 parsing in moses. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 388–394, 2012.

Philip Williams, Rico Sennrich, Maria Nădejde, Matthias Huck, Eva Hasler, and Philipp Koehn. Edinburgh's syntax-based systems at wmt 2014. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 207–214, Baltimore, Maryland, USA, June 2014.

Philip Williams, Rico Sennrich, Maria Nădejde, Matthias Huck, and Philipp Koehn. Edinburgh's syntax-based systems at wmt 2015. In *Proceedings of the EMNLP 2015 Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, September 2015.

Philip Williams, Rico Sennrich, Maria Nădejde, Matthias Huck, Barry Haddow, and Ondřej Bojar. Edinburgh's statistical machine translation systems for wmt16. In *Proceedings of the First Conference on Machine Translation*, pages 399–410, Berlin, Germany, August 2016. Association for Computational Linguistics.

Dekai Wu and Pascale Fung. Semantic roles for SMT: a hybrid two-pass model. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 13–16, 2009a.

Dekai Wu and Pascale Fung. Can semantic role labeling improve SMT? In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pages 218–225, 2009b.