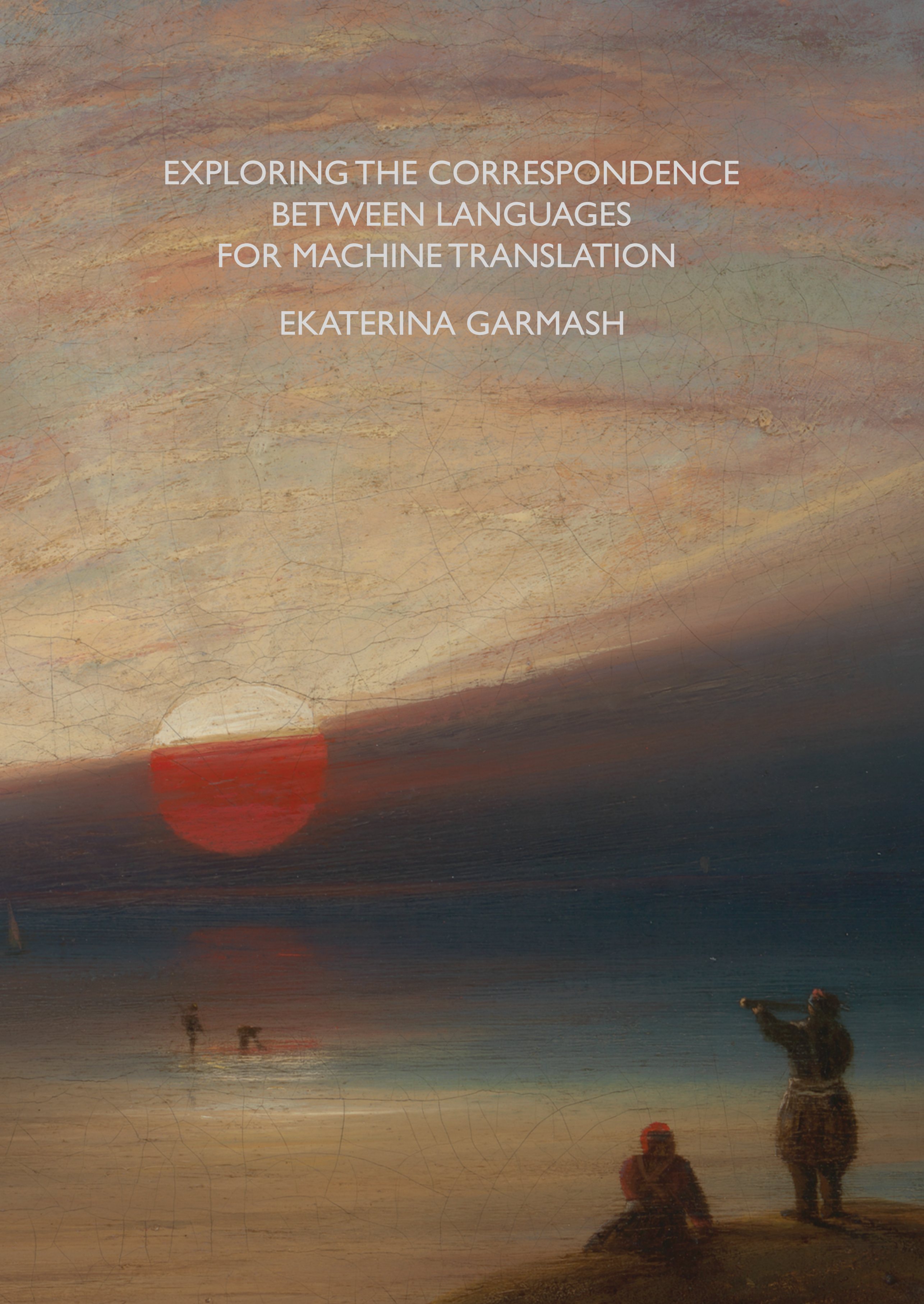
EXPLORING THE CORRESPONDENCE BETWEEN LANGUAGES FOR MACHINE TRANSLATION

EKATERINA GARMASH



Exploring the Correspondence between Languages for Machine Translation

Ekaterina Garmash

Exploring the Correspondence between Languages for Machine Translation

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel op
dinsdag 12 december 2017, te 10:00 uur

door

Ekaterina Garmash

geboren te Kiev, Oekraïne

Promotiecommissie

Promotor:

	Prof. dr. M. de Rijke	Universiteit van Amsterdam

Co-promotor:

	Dr. C. Monz	Universiteit van Amsterdam

Overige leden:

	Prof. dr. J. Bos	Rijksuniversiteit Groningen
	Dr. E. Gavves	Universiteit van Amsterdam
	Dr. E. Kanoulas	Universiteit van Amsterdam
	Prof. dr. K. Sima'an	Universiteit van Amsterdam
	Prof. dr. F. Yvon	Université Paris Sud

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 639.022.213.

Cover by Ekaterina Garmash and Christophe Van Gysel. A fragment of "Sunset on the Beach at Sark" by unknown artist was used (Yale Center for British Art, Paul Mellon Collection).
Printed by Off Page, Amsterdam

ISBN: 978-94-6182-853-8

Acknowledgements

My biggest thanks go to my supervisor Christof. Throughout the four years, he gave me quite enough freedom to choose the topics to work on, however uncertain or diverse they were. Naturally, this ended up in failure a few times, but this was a great way to understand what I could do and what I still had to learn. Of course at the same time, Christof has taught me a lot about how to do research, how to write academic papers and give presentations.

I would like to thank Maarten for his extremely valuable feedback during the last stages of thesis writing. In addition, I would like to thank him for leading the ILPS group, a very big and cool group of people, which I was happy to be a part of.

Lots of thanks to François, Johan, Efstratios, Evangelos and Khalil for being on my committee and therefore agreeing to carefully read this thesis. I also would like to thank Khalil for being my first supervisor in machine translation during my time at ILLC, together with Gideon Maillette de Buy Wenniger. They helped me get into MT and without them none of this would have happened, perhaps.

Beside my promotors and committee, a few more people directly helped me finalize my PhD. I thank Christophe for translating my thesis summary into Dutch and helping with the book cover, my paranymphs Marzieh and Riccardo for agreeing to sit next to me at the defense and help with the organizational matters and Nikos for agreeing to be a substitute paranymph.

Next, I want to thank the people of ILPS. They were admirable researchers and great people. I was often inspired by their knowledge and intelligence and engineering skills. I also knew I could always turn to them for advice or entertainment. Some of them even became my friends. Here is the traditionally exhaustive list of everyone I got a chance to be colleagues with: Abdo, Adith, Aldo, Aleksandr, Alexey, Ana, Anne, Anya, Arianna, Artem, Bob, Boris, Chang, Christof, Christophe, Chuan, Cristina, Daan, Damien, Dan, Dat, David, David, Dilek, Eva, Evangelos, Evgeny, Fei, Hamid, Harrie, Hendra, Hendrik, Hendrike, Hosein, Ilya, Isaac, Iva, Ivan, Julia, Kaspar, Katja, Ke, Lars, Maarten, Maarten, Manos, Marc, Marlies, Marzieh, Masrour, Mostafa, Nikos, Petra, Praveen, Richard, Ridho, Rolf, Shangsong, Spyros, Tatiana, Thorsten, Tobias, Tom, Wouter, Xiaojuan, Xinyi, Yaser, Zhaochun and Ziming.

My PhD studies was not all ILPS. I was fortunate to spend a few summer internship months at some wonderful companies and locations. I thank Artem Sokolov from Amazon in Berlin and Jon Clark from Microsoft Research in Redmond for inviting and mentoring me.

Finally, following the custom, I'd like thank some people who were not directly involved in my PhD but had a profound effect on me nevertheless. Among my close acquaintances Riccardo and Tong were the closest. Strictly speaking, Christophe does not fit into this list as he was a colleage but he has been a great boyfriend as well. My family was the least involved in my PhD work but their influence and support have been unmeasurable.

Katya Garmash
Moscow, October 2017

Contents

1 Introduction

Machine translation (**MT**) is a field of natural language processing that investigates methods to automatically translate texts from one language into another. The first major point of categorization of MT frameworks is between *rule-based* (Nirenburg, 1989) and *data-driven* frameworks (Brown et al., 1993; Koehn, 2009; Carl et al., 2004). Rule-based systems are in a nutshell a set of rewrite rules specifying how to transform an input sequence to an output sequence. The rules are designed by human experts. On the other hand, data-driven approaches derive a way to translate one language to another by 'observing' and learning patterns of translation correspondence from data. The underlying assumption of this method is that a learning algorithm is universal and can be applied to any language or language pair, provided sufficient training data is available. Data in MT typically comes in the form of *parallel corpora*, also referred to as *bitexts*, which are tuples of text conveying the same information in two or more languages (Brown et al., 1993). In the standard case, parallel corpora are bilingual, and the language from which one translates is called the *source* language, while the language one translates into is called the *target* language. Many available parallel corpora are sentence-aligned. Another source of data are *comparable corpora*, where the texts in each language convey *approximately* the same information (Munteanu and Marcu, 2002).

Naturally, there are also hybrid systems, combining elements of both data-driven and rule-based systems.[1] In this thesis we work solely within data-driven frameworks, and more specifically *statistical* frameworks, where one uses principles of statistical learning to obtain the optimal translation procedure. The core problem areas of the statistical approach to MT are instances of many other machine learning applications: model estimation and optimization, approximate inference (since exhaustive search is usually intractable), data selection and generation, output evaluation, model transfer and domain adaptation, and system combination. The two major state-of-the-art frameworks of the statistical approach are *statistical machine translation* (**SMT** (Koehn, 2009)) and *neural machine translation* (**NMT** (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014)).[2] In this thesis we work with both types of frameworks.

[1] Strictly speaking, almost any data-driven system has some elements of a rule-based approach in the pipeline, for example pre- or post-processing rules.

[2] Even though NMT does not have the word "statistical" in its name, it is still based on methods from statistical learning. On the other hand, SMT has this word in its name because it was the first major statistical framework in MT (Brown et al., 1993).

SMT and NMT are different in the way they conceptualize the translation task, which entails differences in the approaches to research areas listed above. SMT decomposes the translation correspondence between sentences and longer strings into a correspondence between smaller units, such as words or sequences of words. In particular, phrase-based SMT (**PBSMT** (Koehn et al., 2003)) operates with phrase pairs, where a phrase is a contiguous string of words (n-gram). Syntax-based SMT (Wu, 1997b; Yamada and Knight, 2001) operates with pairs of simple tree structures. The problem of translation thus consists of estimating a translation distribution for such basic units (translation probability estimation) and learning a model for deciding in which order to translate parts of an input text (reordering problem). SMT is characterized by the modularity of its architecture, where each of the modules (commonly called features) can be trained independently. NMT defines the problem of translating one sentence into another as an end-to-end task and does not attempt to learn an independent model of more local translation correspondence. Unlike PBSMT (and other flavors of SMT), in its current form NMT is characterized by the relative simplicity of its model architecture, and most of the burden of system training lies on finding the right estimation method and having sufficient amounts of training data.

This thesis addresses a diverse set of questions that are united by the general quest for understanding and exploiting structural differences between languages to enhance machine translation quality. In the first part of the thesis, we focus on how syntactic representations can be used to characterize the translational correspondence. Our work is grounded in the assumption that the syntactic structure of both source and target languages is similar enough for re-using language-specific syntactic representations to define features for a translation system. In the second part of the thesis we take an opposite stance and exploit the differences between languages. The main idea here is to employ several language pairs when translating. The benefit is that different translation systems for different language pairs make a diverse set of predictions, but also that the resulting aggregate evaluation of translation hypotheses is typically more accurate.

The two parts of the thesis are also split according to the base machine translation framework that we build upon. The first part is about phrase-based statistical machine translation (PBSMT) and looks into structural regularities of translation correspondence of *subsentential* units. The second part is about neural machine translation (NMT), namely the sequence-to-sequence class of models, and looks how we combine *outputs* of multiple translation systems.

1.1 Research outline and questions

We now start with formulating the central research questions addressed in this thesis. The research questions for the first part of the thesis revolve around the universality of syntactic structures and their role across languages. Syntactic representations have been used extensively in machine translation. Some of their major applications include using syntax to define bilingual structures to better characterize the translation process and to constrain search (decoding). Some approaches to MT have syntactic structures at the core of their framework (such as syntax-based MT (Yamada and Knight, 2001; Quirk and Menezes, 2006; Huang et al., 2006; Shieber, 2007; Liu et al., 2007; Shen et al., 2008)), some use syntactic models and constraints as additional features Ge (2010); Xiang et al. (2011); Lerner and Petrov (2013). One important point of categorization of syntactic methods in MT is whether the training of a system starts with standard word alignment (IBM models (Brown et al., 1993)) and is constrained by syntax at a later stage, or whether it uses syntactic representation to find atomic translation correspondences (Eisner, 2003; Ding and Palmer, 2004; Gildea, 2004). Most methods fall into the first group, and so do ours. Specifically, we assume that syntactic relations between words are preserved when mapped through word alignments, or at least mapped in a systematic way.

Both contributions in Part I use syntax as an additional feature incorporated in a phrase-based system, which is a syntax-agnostic framework and models translation as a flat sequential process. Such an approach allows one to study the role of syntax in a relatively isolated way, as opposed to syntax-based MT. On a high level, in Part I we define language models of bilingual parallel sentences based on syntactic representations. We refer to such language models as *bilingual language models*, as they characterize sequences obtained in a bilingual process (translation). An important aspect of our general approach in this part of the thesis is the move towards simplification of the original structural sentential representations. Many syntactically augmented methods model translation of a source sentence as a restructuring of its parse tree, guided by the complex constraints of the tree formalisms (Wu, 1997a; Huang et al., 2006; Lerner and Petrov, 2013). In the same way, one can model translation as the building up of a target tree (Yamada and Knight, 2001). On the contrary, we stay in the realm of sequential processes.

Our first method to employ syntax is to use tree-based representations of basic bilingual units that are operated on during the process of translation. Phrase pairs are commonly used basic bilingual units. However, we choose to work with bilingual language model tokens (see Section 4.2.1 for a motivation). The tree-based representations are obtained from the complete syntactic parses of the source sentence. Our approach can be seen as approximating the tree construction by generating a sequence of uniformly structured tree fragments. We design this method to specifically improve reordering, i.e., the order in which words and phrases are translated and added to the right-hand side of the partial translation hypothesis. This brings us to our first research question:

RQ1 Can we improve reordering by modeling sequences of syntactic structures representing basic operational units of translation?

RQ1.a. Can the representations only include the local syntactic information of a node in a syntactic parse? What is the minimum context that the local representation should incorporate?

RQ1.b. How do local syntactic representations compare to representations including explicit lexical information of the basic translational units?

RQ1.c. What kind of reordering phenomena are captured by such models?

The second model presented in Part I is built to derive global sentential syntactic structure as a result of translation. Namely, the method consists in building up a parse of a translation hypothesis, with the constraint that all the decisions about the parse operations are strictly guided by the structure of the source sentence and the order in which the target sentence was generated. The obtained syntactic structure of the target sentences is used as input to a syntactic language model, a common class of language models used in natural language processing. The target parse in our method is the result of the translation sequential process, and not a separate probabilistic process modeling a parse tree generation. The question is, how meaningful the parse obtained by projection is and how useful it is for improving translation (when fed to a syntactic language model). We do not focus on a particular aspect of translation (like reordering) but rather investigate how justified it is to characterize the translation process between two languages in terms of structures produced just for one of the languages:

RQ2 Is there a systematic mapping between source and target syntactic representations in a parallel sentence and can it be used to improve translation?

RQ2.a. Is there a universal characterization of a mapping between source and target structure? Can this characterization be used to constrain the decoding process to produce better translations?

RQ2.b. Can the mapping be defined in terms of projection constraints between *elementary* parts of source and target structures? Can we fit a statistical model over the resulting corresponding source and target structures to characterize the overall mapping?

RQ2.c. What are the important mapping constraints that result in structured language models improving translation output?

For **RQ2**, we compare two basic approaches to answering it. The first one, outlined by **RQ2.a**, assumes we can design some characterization that can be directly imposed during the search of a translation hypothesis. The second one, **RQ2.b**, proposes to start with a set of constraints on how elementary substructures should be mapped during translation to obtain a corresponding target structure. The conceptual difference between the approaches from **RQ2.a** and **RQ2.b** is that the latter can learn a characterization (in terms of the parameters of the statistical model) of an *arbitrary* mapping, while the former requires one to explicitly predefine the mapping. We exploit the advantage of the latter approach by defining mappings in terms of different kinds of constraints on the correspondence between elementary structures. Different combinations of such constraints will produce different mappings. **RQ2.c** is about which of the constraints are important and beneficial for translation.

We formulated the research questions from Part I based on the idea that there are systematic similarities between languages expressible with some syntactic formalisms. Assumptions about such linguistic constraints limit the space of learnable parameters. If the assumptions turn out to be empirically valid, then it facilitates the learning task and thus indirectly leads to more accurate models, which is what we aim for in the end.

At the same time, variable parameters across languages can also be useful for practical purposes. If we fix some semantics and let it be expressed in different languages, then the task of translating from each of the given languages into the same language is likely to be of different degrees of difficulty. This is because some language pairs share a lot of characteristics, which facilitates the training task of finding correspondences between languages, and some do not. But the degree of translation difficulty is not the only way one can distinguish between language pairs. Different pairs of languages will share different common properties. If we fix a target language and consider multiple source languages and the corresponding translation systems, then there is likely to be some aspect of the target language that is better captured by only one of the trained translation systems.

Previous research on system combination for machine translation (Och and Ney, 2001; Matusov et al., 2006; Schwartz, 2008; Schroeder et al., 2009) has exploited this assumption. Most of the existing approaches do not so much focus on some specific cross-linguistic properties that are captured differently by different pairs, but rather propose direct methods to combine the expertise of different translation systems.

In this thesis, we apply these ideas in the new setting of neural machine translation (NMT), in particular within the sequence-to-sequence class of models (Sutskever et al., 2014). Unlike the pre-neural translation frameworks, NMT treats the problem of translation at the level of sentences (sequences) and does not explicitly estimate translation correspondence between smaller linguistic units (words or phrases). Such a "black box" approach to translation is very much in the spirit of the previous research on system combination in machine translation, which does not modify the internal algorithm of decoding or translation correspondence model estimation, but typically considers combinations of completely generated translation hypotheses. Another important aspect of our proposed method of multi-source translation is that it can be cast naturally as a general machine learning problem of ensemble combination. Sequence-to-sequence models are essentially sequential classifiers, so the existing ensemble methods can be easily applied without the additional engineering requirements of pre-neural system combination methods. This allows us to formulate our last research question:

RQ3 Can we exploit the variation in cross-lingual correspondence and improve translation quality with multi-source NMT ensembles?

- RQ3.a. How does ensemble performance depend on the quality of individual translation systems that are part of it?
- RQ3.b. Is there systematicity in what a neural translation system for a given language pair is good at, and what aspects of the target side it reproduces suboptimally? Can we exploit this systematicity in multi-source translation?
- RQ3.c. How do multi-source ensembles compare to ensembles of NMT systems for single language pair trained with different initialization seeds?

We answer **RQ3.a** and **RQ3.b** by designing corresponding translation experiments. For **RQ3.b**, we propose two methods for learning a combination function for ensemble prediction, and evaluate them in translation. We evaluate both multi-source ensembles and monolingual ensembles (produced by different initialization seeds) for both of the research questions. Finally, we answer **RQ3.c** based on the observed performance of the two kinds of ensembles for all types of experiments.

1.2 Main contributions

Here we summarize the main contributions of this thesis. We categorize them into theoretical (new ideas and concepts, new approach to an existing problem), algorithmic (proposing a new method to modify the baseline translation pipeline) and empirical (experimental evaluation of the proposed methods).

1.2.1 Theoretical contributions

1. We propose to generalize bilingual language model tokens (Niehues et al., 2011), which are typically considered as lexicalized forms, and consider syntactic-based representations of arbitrary levels of complexity. This approach can be viewed as an approximate model of restructuring of a source tree during translation (reordering). [**Chapter 4**]

2. We connect the ideas of a direct correspondence assumption (Hwa et al., 2002) between source and target syntactic structures with structured language models (Chelba and Jelinek, 2000) to obtain a simple method to integrate the latter models into phrase-based machine translation. Additionally, we propose to view our bilingual structured language model from the perspective of research on discovering patterns of source and target syntactic correspondence. [**Chapter 5**]

3. We introduce a source of ensemble diversity specific to machine translation where we exploit different source languages while translating into the same target language. [**Chapter 7**]

1.2.2 Algorithmic contributions

1. We propose a method to construct dependency based representations for bilingual language models incorporating the local syntactic context of a words comprising the token. [**Chapter 4**]

2. We describe an algorithm for an incremental projection of syntactic structure via word alignment, which is easily integrated into a phrase-based decoder. [**Chapter 5**]

3. We propose a mixture of experts model for neural machine translation. The core novelty is to regard concatenated hidden states of neural systems in the ensembles as the input to a mixing gate. [**Chapter 7**]

1.2.3 Empirical contributions

1. We evaluate translation performance of a phrase-based statistical machine translation system augmented with bilingual language models with dependency based representations. We evaluate general translation performance, as well as the reordering aspect specifically. We compare our models to the performance of previously proposed bilingual language models. [**Chapter 4**]

2. We conduct an empirical evaluation of structured language models with word alignment-projected parses by integrating it into a phrase-based machine translation scenario as an n-best translation reranking model and as a feature in the decoder. [**Chapter 5**]

3. We evaluate a series of combination methods for multi-source neural machine translation systems for a diverse set of language pairs. We compare the resulting translation quality to the one obtained by monolingual ensembles which are induced by different random parameter initializations. [**Chapter 7**]

1.3 Thesis overview

As outlined in the preceding sections, this thesis revolves around the question of how different or similar languages are and how this diversity affects machine translation. We start with a general background chapter (**Chapter 2 - General Background**) that introduces the core concepts and terminology of statistical machine translation and neural machine translation, and also provides a detailed explanation of the automatic evaluation metrics that we use in our experiments. We organize the thesis into two parts, the first part exploring the idea of inherent syntactic similarity between languages and the second part utilizing the diverse behaviors of different language pairs in translation. In addition to that, the proposed models are grounded in different translation frameworks in the two parts.

- **Part I - Syntax-based Bilingual Language Models for Statistical Machine Translation**. In this part we explore the idea of similarity between syntactic structures in a pair of parallel source and target sentences. We implement our ideas in the form of bilingual language models that incorporate syntactic structure. These language models are used as features in a *phrase-based* statistical machine translation system.

- **Chapter 3 - Background: Concepts, Related Work, Baseline**. This is a background chapter. It provides an overview of syntactic models that have been used to improve phrase-based SMT. We also provide some background on syntactic formalisms commonly used in machine translation. The syntactic model proposed later in Chapter 4 is a so-called bilingual language model, and we also provide the definitions and discuss previous research on this type of language models. Finally, this chapter also contains a detailed specification of the phrase-based system that we use in the experiments in this part of the thesis. We also describe the training and test data.

- **Chapter 4 - Dependency-Based Bilingual Language Models for Reordering in Statistical Machine Translation** is a research chapter where we propose a bilingual language model with tokens based on source syntax. The model is grounded in the idea that source syntax can provide useful information about the reordering process between the source and target sides during translation.

- **Chapter 5 - Bilingual Structured Language Models for Statistical Machine Translation** is a research chapter where we propose a method to adapt structured language models to a bilingual scenario. We use it as a target language model, but instead of having a probabilistic model to infer the target parse (which is part of the structured language model), we propose to deterministically project it from the source sentence via word alignments.
- **Part II - Exploring Diversity in Neural Machine Translation**. The field of machine translation has accumulated an extensive body of observations implying that different language pairs demonstrate substantial differences in translation performance while being trained on the same amount of data and with the same algorithm. We propose to utilize this naturally occurring diversity to the benefit of translation. Building on the foundation of general research on ensemble prediction, we propose a series of models for multi-source neural machine translation ensembles.
- **Chapter 6 - Background: Ensembles, System Combinations, and Baselines.** This chapter provides the relevant background on previous research on ensemble prediction. Additionally, it provides an overview of a conceptually close approach of system combination in (pre-neural) machine translation. Finally, it also provides the specifics of our neural machine translation system and the training and test data that we use in the experiments in Chapter 7.
- **Chapter 7 - Ensemble Learning for Multi-Source Neural Machine Translation** is a research chapter. It proposes to use ensembles of translation systems with different source languages and the same target language during decoding, thus requiring the availability of a multi-parallel test set, but not training set. It introduces a series of combination models used in ensembling.
- **Chapter 8 - Conclusions** summarizes both parts of the thesis and revisits the research questions introduced in Section 1.1. It also provides an outlook for future work.

1.4 Origins

Research presented in the following chapters was based on previosly published papers:

- Garmash and Monz (2014): Ekaterina Garmash and Christof Monz. Dependency-based Bilingual Language Models for Reordering in Statistical Machine Translation. EMNLP, Doha, Qatar, 2014 (**Chapter 4**).

  The model was proposed by Monz. Experiments and analyses were performed by Garmash. Both authors contributed to the text. Garmash did most of the writing.

- Garmash and Monz (2015): Ekaterina Garmash and Christof Monz. Bilingual Structured Language Models for Statistical Machine Translation. EMNLP, Lisbon, Portugal, 2015 (**Chapter 5**).

The model was developed by Garmash. Experiments and analyses were performed by Garmash. Both authors contributed to the text. Garmash did most of the writing.

- Garmash and Monz (2016): Ekaterina Garmash and Christof Monz. Ensemble learning for Multi-Source Neural Machine Translation. Coling, Osaka, Japan, 2016 (**Chapter 7**).

  The models were developed by Garmash. Experiments and analyses were performed by Garmash. Both authors contributed to the text. Garmash did most of the writing.

2 General Background

In this chapter we provide some general background on machine translation (MT). Specifically, we discuss the core concepts and models of statistical machine translation (SMT; Section 2.1), neural machine translation (NMT; Section 2.2), and the evaluation of machine translation output (Section 2.3). In this chapter we intend to provide minimal background for non-specialists in the field to be able to understand the rest of the thesis. Sources for a more thorough introduction include (Koehn, 2009; Goldberg, 2017) and the references throughout this chapter.

2.1 Basics of statistical machine translation

Brown et al. (1993) have laid the foundation for the state-of-the-art SMT. They formalize the task of translating a given foreign sentence F into the target sentence E as finding $\text{argmax}_E p(E|F)$. This formula can be rewritten as:

$$\begin{aligned} E^* &= \text{argmax}_E p(E|F) = \text{argmax}_E \tfrac{p(F|E)p(E)}{p(F)} && (2.1) \\ &= \text{argmax}_E p(F|E)p(E) && (2.2) \end{aligned}$$

The resulting model is referred to as the *noisy channel* model of SMT and defines the process of generating a parallel sentence $\langle E, F\rangle$ as first generating E and then generating E conditioned on F. $p(F|E)$ models translation correspondence itself, while $p(E)$ is called a (target) *language model*.

As we mentioned in Chapter 1, the translation correspondence in SMT is conceptualized in terms of correspondence between minimal units, such as words or multi-word expressions. The translation correspondence between sentences or larger chunks is derived from a correspondence of the smaller units comprising them. Brown et al. (1993) proposed the first statistical model of translation correspondence between individual words, called the *IBM models*. Their paper formalizes the process of translating from a given foreign sentence as first deciding, for each word f in a foreign sentence, what its 'English' translation e is, and then deciding into what position e should be realized. The first step models non-positional translation correspondence and is an instantiation of a *translation model*, the second step models *reordering*. Extensions of this basic model add additional steps to the generative process (see Section 2.1.1).

Later as SMT progressed, a discriminative view on modeling translation was adopted, whereby the conditional probability of translating E from F is modeled as a log-linear function of an arbitrary set of features that are useful for describing the translation correspondence between the two given languages (Och and Ney, 2002):

$$p(E|F) = \frac{\exp\sum_i \lambda_i h_i(E, F)}{Z(F)}, \tag{2.3}$$

where h_i are feature functions each weighted by $\lambda_i s$, and $Z(F)$ is the partition function (which is in practice irrelevant as the translation search problem has the F variable fixed, see Section 2.1.7). Despite the switch of the modeling framework, target language models, translation models and reordering models remain at the core of SMT (comprising the so-called standard model of phrase-based SMT, Section 2.1.2).

In the subsections below we describe the introduced models in some more detail. Section 2.1.1 provides background on the IBM models. Section 2.1.2 explains the basics of the phrase-based model, which follows the overall strategy introduced by Brown et al. (1993), but operates with larger translational units. Sections 2.1.6 and 2.1.7 explain how decoding (inference) and estimation of the parameters of the log-linear model (Equation 2.3) are realized.

2.1.1 IBM models

A central concept of the IBM models is the *alignment* function: a one-to-many map from words (positions) of a foreign sentence F into words (positions) of an 'English' sentence E. Given a corpus of parallel sentences, F and E are observed variables, while alignment a is hidden. Brown et al. (1993) derive a method to estimate distributions of these variables. Namely, IBM models 1 and 2 estimate translation probability $t(e|f)$ between words and alignment probability p_a, which is an instantiation of a reordering model:

$$p_{\text{IBM1,2}}(E, a|F) = \prod_{j=1}^{l_E} t(e_j|f_{a_j}) p_a(a(j)|j, l_E, f_F). \tag{2.4}$$

Brown et al. (1993) estimate these functions via an expectation-maximization algorithm by first deriving an exact expectation of alignment counts, and then performing the maximization step. IBM3 refines IBM2 by adding an initial step of deciding how many English positions a foreign word translates into. This is called a fertility model, resulting in the overall joint probability:

$$p_{\text{IBM3}}(E, a|F) = \binom{l_E - \phi_0}{\phi_0} p_1^{\phi_0} p_o^{l_e - 2\phi_0} \times p_{\text{IBM1,2}}(E, a|F), \tag{2.5}$$

where ϕ_i is the number of words generated by the ith source foreign word and n is the conditional probability over this variable. The model is additionally equipped with a step of generating a NULL token (to model unaligned target words), which is parametrized by ϕ_0, the number of unaligned words, p_0, the probability that no NULL was generated, and p_1, the probability of generating NULL. The new formulation of the model prevents one from deriving efficient exact expectations, which is why the E-step is approximated by sampling.

Another difference of IBM3 from IBM1, 2 is that p_a is modeled as *relative distortion* with respect to the previously placed cept, a tuple of an foreign word the set of English words aligned to it. IBM4 elaborates the relative distortion model by conditioning it on word classes of words in a cept. IBM5 further reformulates the distortion to address the deficiency of the preceding model variants which allow multiple English words to be placed in the same position.

Most SMT models use the IBM models as a preprocessing step to obtain *word alignments* of a parallel corpus. They serve as constraints to extract translation correspondence between larger chunks, such as phrases (Section 2.1.2). While IBM models learn a one-to-many alignment, it is common to run training in both directions (source to target and target to source) to obtain a many-to-many mapping by taking the union of the two alignments excluding alignment pairs which are not in the intersection and are not adjacent to pairs which are in the intersection (the *grow-diag-final* strategy (Och and Ney, 2003b)). GIZA++[1] (Och and Ney, 2003a) is a popular software package for training IBM models and obtaining a word-aligned corpus.

2.1.2 Phrase-based statistical machine translation

Modeling the distributions over single words simplifies what we know about language. Namely, it has been frequently observed that the semantics and the usage of words strongly depend on the context in which they occur such as a sentence or document. Translational correspondence is, to a large extent, semantic correspondence across languages, therefore it could be useful to have a model that can infer the semantics and the translation of a words based on the context they occur in.

Phrase-based SMT (PBSMT (Koehn et al., 2003)) implements this idea by directly considering words in their context when modeling translation correspondence. The framework defines *phrases* as contiguous strings of words observed in a sentence. Given a preprocessed word-aligned corpus, a *phrase pair* is defined as a pair of observed source and target phrases such that no word in either of the phrase is aligned to a word outside of this pair. With this definition, phrases in a parallel sentence can be ordered hierarchically. Typically, phrase lengths are limited to 5–7 words (Koehn et al., 2003). A minimal translational unit in PBSMT is a minimal phrase pair, i.e., one that cannot be further decomposed into sub-phrase pairs.

As we mentioned above, the problem of estimating translation probabilities and the problem of reordering and language modeling are at the core of SMT modeling. Namely, they are typically used as features in the log-linear interpolation function of translation correspondence between languages (Equation 2.3). Below we provide background on concrete model instantiations of these problems which are commonly used in PBSMT systems (Section 2.1.3-2.1.5). Following Koehn (2009), we refer to a system consisting of these models as the *standard model of PBSMT*.

Besides modeling the source or target sentence and their correspondence, an important question is how to find the optimal translation E^*, given a source sentence F and a trained model $p(E|F)$ (Equation 2.3), a task called *decoding*. Since a phrase pair is the

[1] `http://www.statmt.org/moses/giza/GIZA++.html`

basic structure of PBSMT, the task of decoding is formulated as finding the *derivation* of E^*, which is an ordered sequence of phrase pair applications:

$$\textit{derivation} = \langle \overline{e}_1, \overline{f}_1 \rangle, ..., \langle \overline{e}_k, \overline{f}_k \rangle,^2 \tag{2.6}$$

so that in the end there is no phrase in the input sentence that is not translated ($\overline{f}_1, ..., \overline{f}_k$ in Equation 2.6 cover the whole input sentence, but not necessarily in this order), and no phrase is translated twice (the intersection of $\overline{f}_1, ..., \overline{f}_k$ is empty). This is implemented by keeping track of a *coverage vector* during decoding. The concatenation of $\overline{e}_1, ..., \overline{e}_k$ (in this order) is the output translation hypothesis. Given this formulation of decoding, it is an NP-complete problem (Knight, 1999) and one has to resort to approximations. In Section 2.1.6 we describe the most common version of the PBSMT decoding algorithm. Additionally, we describe a set of features, which are also part of the PBSMT standard model, that directly characterize the decoding process.

Beside estimating the models included into the log-linear interpolation, the feature weights themselves have to be tuned. This is done on a held-out set, by sampling from $p(E|F)$ (via the given decoding algorithm). The common approaches to tuning are outlined in Section 2.1.7.

2.1.3 Translation models

For each phrase pair $\langle \overline{e}, \overline{f} \rangle$ extracted from a given word-aligned corpus (see Section 2.1.2), the standard PBSMT model estimates four translation probabilities: conditional phrase translation probabilities in both directions ($p(\overline{e}|\overline{f})$ and $p(\overline{f}|\overline{e})$), and lexical translation probabilities in both direction ($p_{lex}(\overline{e}|\overline{f})$ and $p_{lex}(\overline{f}|\overline{e})$). The conditional translation probabilities are estimated by their relative counts (Equation 2.7, analogously for $p(\overline{f}|\overline{e})$). The use of lexical translation probabilities is motivated by the need of alternative estimation methods for infrequent phrase pairs for which simple counts may be unreliable. They are computed by looking into the phrase-internal alignment and taking the normalized product of word translation probabilities of the alignment pairs inside the phrase (Equation 2.8, analogously for $p_{lex}(\overline{f}|\overline{e})$). In addition, it is also possible to smooth these maximum-likelihood estimates (Foster et al., 2006):

$$p(\overline{e}|\overline{f}) = \frac{\textit{count}(\langle \overline{e}, \overline{f} \rangle)}{\sum\limits_{\overline{f}' \in \textit{source corpus}} \textit{count}(\langle \overline{e}, \overline{f}' \rangle)}, \tag{2.7}$$

where $count$ returns the number of occurences of a phrase pair in word-aligned parallel corpus. And:

$$p_{lex}(\overline{e}|\overline{f}) = \max_{a \in \textit{observed alignments of} \langle \overline{e}, \overline{f} \rangle} \prod_{i=1}^{\textit{length}(\overline{e})} \frac{1}{|j|(i,j) \in a|} \sum_{(i,j) \in a} w(e_i|f_j), \tag{2.8}$$

[2] Note that the index here denotes the order of a phrase pair application, not the position of phrases in a sentence.

where $w(e_i|f_j)$ is the IBM word translation probability defined analogously to $p(\overline{e}|\overline{f})$. The max operator in Equation 2.8 can be substituted by a sum operator, or by an operator selecting the most frequent alignment for the given phrase pair.

2.1.4 Language models

The most common language model (LM) used in MT is the n-gram language model. It models a sequence in a natural language (sentence) as a product of probabilities of each word in a sequence conditioned on the history of this word in this sequence. An LM of order n conditions each word on a sequence of $n-1$ immediately preceding words:

$$p(w_1, ..., w_m) = \prod_{i=1}^{m} p(p_i|p_{i-n+1}...p_{i-1}). \tag{2.9}$$

Typically, the smallest order of n-gram models in SMT is 3.

In count-based models,[3] conditional probabilities are estimated by relative counts on a large monolingual corpus. Since monolingual corpora are easier to obtain than parallel corpora, LMs can be trained on substantially larger data sets, which results in more accurate estimates, potentially compensating for some of the shortcomings of the translation model. In addition to plain relative counts, smoothing techniques are used to obtain more realistic estimates and to deal with the zero observation problem during testing (Goodman, 2001). Common smoothing techniques include Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1996) and Witten-Bell smoothing (Witten and Bell, 1991).

2.1.5 Reordering models

The reordering problem of PBSMT is about how to model the sequence of phrase pair applications that produces the output translations. The observed diversity of reordering patterns across language pairs is substantial, which has lead to a great variety of models proposed over the years (Bisazza and Federico, 2016), which includes our own contribution that models reordering as a sequential process (Chapter 4, (Garmash and Monz, 2014)). However, the standard models includes the relatively simple, but effective *linear distortion* and the *lexicalized distortion model*.

Linear distortion (Koehn et al., 2003) is a model of the distance (in word positions) between start of the foreign phrase translated at the phrase pair application step i and the end of the phrase translated at step $i-1$.

The lexicalized distortion model (LDM (Tillmann, 2004)) is a set of conditional distributions of *orientations* given the phrase pair that is applied during translation. The orientation is a variable denoting the relative order of source sides of phrase pairs applied at times i and $i+1$ ($\langle \overline{e}_i, \overline{f}_i \rangle$ and $\langle \overline{e}_{i+1}, \overline{f}_{i+1} \rangle$).[4] One distinguishes between monotone, swap and discontinuous orientations. *Monotone* orientation means that the source side of the phrase pair applied at time step $i+1$ is adjacent to the right of the source side of

[3] I.e., not neural models (Bengio et al., 2003), which have also been used in SMT (Luong et al., 2015b).

[4] Note that the index i here denotes the order in which a phrase pair is used in a derivation, not the position of the phrases in a sentence.

the phrase pair from time step i (*position*$(\overline{f}_{i+1})-$*position*$(\overline{f}_i) = 1$).[5] *Swap* orientation means that is it adjacent to the left (*position*$(\overline{f}_i)$ − *position*$(\overline{f}_{i+1}) = 1$). *Discontinuous* orientation means that the source sides are not adjacent (*abs*(*position*$(\overline{f}_{i+1})$ − *position*$(\overline{f}_i)) > 1$).[6] Typically, each of the orientations are estimated with a separate conditional distribution. The conditioning variable comprises the entries of the phrase table itself. Additionally, each of the distributions is split into two by either considering the current phrase pair as the phrase pair from step i, or $i+1$ (forward-looking and backward-looking distortion, respectively). Thus, the standard model typically has six LDM models.

2.1.6 Decoding

As we mentioned above, exhaustive search is intractable in PBSMT, therefore approximations are used. The most common decoding method is beam search whereby stacks with partial hypotheses are organized by the number of source words translated. Thus, decoding starts from the stack corresponding to zero translating words, containing one hypothesis ($\langle s \rangle$). For each stack, each partial hypothesis is expanded with k top phrase pairs from the phrase table. The resulting partial hypotheses obtained from applications of the candidate phrase pairs are scored with respect to log-linear combination of features and then assigned to a corresponding new stack. The stacks are pruned (with respect to log-linear translation probability) to keep the stack size below the fixed beam size. Specifying the decoding procedure in more detail is beyond the purpose of this chapter, and we refer the reader to (Koehn, 2009).

In addition to the translation modeling features described above (Section 2.1.3–2.1.5), the standard model also includes a set of features directly characterizing decoding. A *word penalty* is a binary feature firing for each new target word added to the partial hypothesis. A *phrase penalty* is a binary feature which fires with each phrase application.

During pruning an additional a *future cost* function is used to heuristically approximate the expected cost of translating the untranslated part of the source sentence, given the partial hypothesis.

In addition to the beam size, another hyper-parameter restricts the search space: *distortion limit*. It is defined as the maximal distance (in word positions) between the previously translated source phrase and the currently translated source phrase. In addition to simply reducing the search space, it also prevents the decoder from exploring unrealistically long reorderings; for instance, English and French typically involve only very short distance reorderings.

2.1.7 Tuning of log-linear weights

The weights of the features in Equation 2.3 are tuned iteratively on a held-out parallel set (tuning/development set) until convergence. The tuning objective is a translation evaluation metric, typically BLEU (see Section 2.3.1). The log-linear model in Equation 2.3

[5] Note that the target sides of the phrase pair from time step $i+1$ is always adjacent to the right of the target side of the phrase pair from step i.

[6] In some versions discontinuous-left and discontinuous-right are distinguished.

is reformulated as a linear model, since:

$$E^* = \operatorname{argmax}_E \frac{\exp \sum_i \lambda_i h_i(E,F)}{Z(F)} \tag{2.10}$$

$$= \operatorname{argmax}_E \sum_i \lambda_i h_i(E,F), \tag{2.11}$$

where $h_i(E,F)$ is interpreted as either a binary feature (Section 2.1.6) or the logarithm of a probability model from Section 2.1.3–2.1.5. The most common algorithms are MERT (Och, 2003), MIRA (Watanabe et al., 2007), and PRO (Hopkins and May, 2011). They involve updating the linear weights λ_i based on the metric scores of hypotheses sampled from a decoder (with feature weights from the previous iteration).

2.2 Neural machine translation

In recent years, neural machine translation has emerged as a major framework of data-driven translation and by now has become, if not state-of-the-art, then at least the focus of academic research in machine translation.

The structure of an SMT system is very modular (Section 2.1), and its end-to-end training typically requires a series of independent training steps, including computation of word alignments, estimation of each model (feature) separately, and tuning of the feature weights. In contrast, the neural framework formulates the task as an end-to-end task of generating a target sequence given an input source sentence with one connected neural network, without decomposing it into individual steps and building blocks like in SMT. On the one hand, such an approach does not allow, like in SMT, to train system modules independently and the combine them later. Yet in SMT combining different models together often requires some additional engineering effort, for example in system combination, see Section 6.2. More importantly, while state-of-the-art performance of the two types of framework are comparable (Bentivogli et al., 2016), the internal architecture of a typical NMT system is considerably simpler than the structure of a phrase-based system. In NMT, quality is improved by utilizing *domain-agnostic* methods from the general research on neural networks, such as: the choice of neural units, number of layers, choice of activation function, regularization, optimization. Such a generic definition of NMT also allows for relatively easy transfer across modalities (Huang et al., 2016; Elliott and Kádár, 2017).

In the following subsections we describe the most commonly used NMT architectures and how inference is carried out (Section 2.2.1), as well as the relevant optimization procedure (Section 2.2.2). We should point out that the research area of NMT is developing very rapidly, and for this reason even the current state-of-the-art models may become obsolete soon. Additionally, several aspects of NMT, such as decoding, have not yet been investigated to the same extent as in SMT. Here we describe models and methods most frequently used as baselines in recent research papers.

2.2.1 Architecture

The high-level structure of a baseline NMT network is commonly referred to as an *encoder-decoder* architecture (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014;

Bahdanau et al., 2015) and most models fall under this type of architecture. An encoder is a network that takes as input the source sentence and returns a real-valued representation intended to capture all the necessary information needed to generate the target sequence. A decoder is a network that, conditioned on the input representation, generates a target sequence.

The simplest classic instantiation of an encoder-decoder model is a sequence-to-sequence model (Sutskever et al., 2014). It defines both the encoder and the decoder as recurrent neural networks (RNNs). Encoding is done by feeding the source sequence to the first RNN element by element.[7] At each decoding step i, the output hidden representation of the decoder RNN h_i^t is mapped to a layer with a size of the target vocabulary, which is transformed via a *softmax* operation into a probability distribution y_i. The problem of predicting a sequence is thus formulated as a sequential element-wise classification problem. To connect this model to the SMT formulation of translation, the sequence-to-sequence model can be described as:

$$p(E|F) = \prod_{i=1}^{n} p(e_i|\text{decoder}(e_{<i}, \text{encoder}(f_1, ..., f_m))), \tag{2.12}$$

where the conditioning decoder$(e_{<i}, \text{encoder}(f_1, ..., f_m))$ is the hidden state of the RNN after processing the preceding words $w_1...w_{i-1}$. The hidden state of the decoder is initialized with a representation output by the encoder, encoder$(f_1, ..., f_m)$.

A second major development within NMT was the introduction of an attention mechanism (Bahdanau et al., 2015). Attention is a network that, conditioned on a hidden state of the decoder at each time step i, computes a probability distribution over the input sequence. This distribution is used to compute a weighted sum over representations of the input words (typically, the hidden states of the encoder RNN). The resulting *context vector* is incorporated into the decoder as another hidden state and the output distribution is produced in the same way as described above. In formal notation, the decoding step consists of the following major steps:

$$y_i = \text{softmax}(W_y \widetilde{h_i}) \tag{2.13}$$

$$\widetilde{h_i} = \tanh(W_c[c_i; h_i^t]) \tag{2.14}$$

$$c_i = \textstyle\sum_{j=1}^{m} \alpha_i[j] \cdot h_j^s \tag{2.15}$$

$$\alpha_i = \text{softmax}(\textit{score}(h_i^t, H^s, c_{i-1})) \tag{2.16}$$

where $[j]$ denotes the operation of indexing a vector at position j, h_i^t is the output of the decoder RNN, $\widetilde{h_i}$ is the final hidden layer after the basic recurrent transformation h_i^t is done, y_i is the target probability distribution at time step i, and context representation c_i is computed, and *score* is the function that computes the distribution over the source sequence, given the tensor H^s which is a concatenation of the source sentence's hidden states $h_1^s, ..., h_m^s$, corresponding to the input sequence $f_1, ..., f_m$. A number of variants for the *score* function have been proposed. The classic one (Bahdanau et al., 2015) is a recurrent network. Non-recurrent attention functions (that do not depend on c_{i-1} in

[7] By default, sequences consist of words, but subword units (Sennrich et al., 2015) or characters (Ling et al., 2015) are also a popular choice.

Equation 2.15) show comparable performance, while still somewhat underperforming with respect to a recurrently defined attention (Luong et al., 2015b).

The attention mechanism in (Bahdanau et al., 2015) was inspired by word alignments in SMT (Section 2.1). Although NMT is defined as an end-to-end task, still intuitively at each decoding step the predicted word corresponds to a specific subset of the input sequence. The attention mechanism defines the distribution of inclusion into this subset. From a computational perspective, even the most advanced recurrent units may not be effective enough to carry over the necessary information until the end of the predicted sequence. The attention mechanism allows one to look back to the original conditioning source sequence at every prediction step. A recent model (Vaswani et al., 2017) pushes this idea even further by abandoning recurrent units altogether and only rely on attention.

A sequence-to-sequence model with attention is usually used as a baseline system, and therefore is also our baseline model of choice in Chapter 7. The current research in NMT architecture revolves around alternative ways of encoding (Ling et al., 2015; Eriguchi et al., 2016; Bastings et al., 2017), modifying the iterative decoder (Vaswani et al., 2017), and integrating additional 'remembering' states into the decoder (Tu et al., 2016; Chen et al., 2017).

Since a decoder is a recurrent model, decoding is implemented as a beam search. So far, most effort in NMT is put into translation modeling and optimization, however, some recent work addresses decoding as well (Wiseman and Rush, 2016; Gu et al., 2017).

2.2.2 Optimization

The most common optimization objective is the negative log-likelihood of the probability distribution defined by the encoder-decoder network (Equation 2.12) estimated over the reference target sequences:

$$\begin{aligned} \mathit{NLL} &= -\sum_{\langle E,F\rangle} \log(p(E|F)) && (2.17)\\ &= -\sum_{\langle E,F\rangle} \log(\prod_{i=1}^{n} p(w_i|\text{decoder}(w_{<i}))) && (2.18)\\ &= -\sum_{\langle E,F\rangle} \sum_{i=1}^{n} \log(p(w_i|\text{decoder}(w_{<i}))) && (2.19) \end{aligned}$$

This factorization allows us to update the parameters of the network for each predicted word in a sequence via stochastic gradient descent (SGD). In practice, gradients are grouped by mini-batches, and often more advanced optimizers are used than simple SGD (Kingma and Ba, 2015).

The described optimization procedure is simple and effective, however it does not target what an NMT system is designed for. Negative log-likelihood measures how 'surprised' the network is by the correct translation sequence. However, a NMT system is intended for generating sequences, without any (target) input. One approach to directly target this goal is by optimizing the objective that the reference translation should be generated by the decoder RNN in a beam search (Wiseman and Rush, 2016). Another approach is to directly optimize the network with respect to some translation quality network, such as BLEU (Papineni et al., 2002) by sampling (Ranzato et al., 2015). Despite these considerations, maximum likelihood estimation with *NLL* remains the most common method in NMT optimization.

2.3 Evaluation

2.3.1 Evaluation metrics and tools

Evaluation of translation output quality is an active research area, and new automatic metrics are regularly proposed and evaluated (Bojar et al., 2016). However, in our work we only use established MT metrics. In addition to that, we use a more specialized metric *LRscore*, which is designed to evaluate reordering (Chapter 4).

Typically, an MT evaluation metric takes as input the translation output and the human reference translation. Some metrics, such as the LRscore, also require precomputed word alignments between source and reference and between source and translation output. Below we describe each of the metrics and refer to the tools that we used to compute them.

BLEU (Papineni et al., 2002) is the most popular automatic metric in MT. It is based on the idea of *non-positional* matching of parts of the system output and a reference. The matching units are *n-grams*, contiguous sequences of n words in a sentence. In a nutshell, an n-gram in a translation output sentence matches if this n-gram also occurs in the corresponding reference sentence. BLEU allows us to use multiple references, in which case the definition of matching is modified (this is called *clipped counts*, see Equation 2.22).

The matching counts are used to define the n-gram precisions (for each n separately), which itself can be used as a separate metric and which we define formally below. The final formula is a product of n precisions multiplied by a brevity penalty, designed as an approximation of recall:

$$\text{BLEU-n} = bp \cdot \prod_{i=1}^{n} prec_i \tag{2.20}$$

$$bp = \sum_{j=1}^{\substack{\#\ candidate \\ sentences}} \min\left(1, \frac{length\ (output_j)}{length\ (ref_j)}\right) \tag{2.21}$$

where *ref* is the reference sentence (from ref_j^1,...,ref_j^m) with the closest length to the translation output sequence.[8]

In our experiments we use an in-house implementation of BLEU-4 and refer to it simply as BLEU.

prec$_n$ (n-order ngram precision). This metric is a component of BLEU, but it can also be used independently as a simple metric of word order accuracy. Of course, this is a crude metric, since it does not abstract away from lexical identity of words, although

[8] In some implementations the shortest reference is picked

this is partially alleviated by allowing for multiple references. In our experiments we use *prec*$_4$.

$$prec_n = \frac{\sum\limits_{j=1}^{\substack{\#\,candidate\\ sentences}} \sum\limits_{n\text{-}gram} count_{clip}(n\text{-}gram, output_j, ref_j^1, ..., ref_j^m)}{\sum\limits_{\substack{\#\,candidate\\ sentences}} \sum\limits_{n\text{-}gram} count(n\text{-}gram, output_j)} \tag{2.22}$$

where $count_{clip}(n\text{-}gram, output, ref^1, ..., ref^m)$ outputs the minimum number of occurrences of *n-gram* among *output*, *ref*^{1},...,*ref*^{m}.

METEOR (Lavie and Denkowski, 2009) computes alignments between the words of the translation output and the reference by iteratively deciding which words match together. Matching is defined as exact matching, stem-based matching, and synonym matching amongst others. The concrete model can vary, but the high-level formula is:

$$\text{METEOR} = (1 - Pen) \cdot \frac{P \cdot R}{\alpha P + (1-\alpha)R} \tag{2.23}$$

$$Pen = \gamma \cdot frag^{\beta}, \tag{2.24}$$

where P and R are precision and recall, respectively, of the translation matching the reference, given the established one-to-one alignment. *frag* is a fragmentation measure. The values of P, R, *frag* are aggregated over the whole test set and then combined into the formula in Equation 2.23. In the case of multiple references, for each sentence, the reference producing the highest score gets chosen. The hyperparameters α, γ, β are tuned on a held-out set against human-generated labels.

We use version 1.4 of the METEOR software (Denkowski and Lavie, 2011),[9] and in particular the metric tuned on the HTER task. We run the METEOR system with the HTER task setting.

TER (Translation Error Rate (Snover et al., 2006)) measures the number of edits required to rewrite a system output into a reference. Specifically, it is designed to compute the minimum number of the following edit operations: insertion, deletion, substitution, and shift. The number of edits is computed by greedy search, and is based on exact lexical matching between words. Since the goal is to find the minimum edit distance, for the case of multiple references *ref*$_1$,...,*ref*$_m$, the one with the smallest number of edits is chosen for each translation output sentence:

$$\text{TER} = \frac{\sum\limits_{j=1}^{\substack{\#\,candidate\\ sentences}} \min_k \#edits(output_j, ref_j^k)}{\sum\limits_{j=1}^{\substack{\#\,candidate\\ sentences}} \frac{\sum_k length(ref_j^k)}{m}} \tag{2.25}$$

[9] http://www.cs.cmu.edu/~alavie/METEOR/download/meteor-1.4.tgz

We use the implementation provided by the authors.[10]

LRscore (Birch and Osborne, 2010) is designed to evaluate reordering. The basic idea behind this metric is to interpret the alignment between the source sentence and the target sentence as a permutation of the source sentence. Then, the permutation of the source (the input sentence) with respect to the reference and the permutation with respect to the translation output are compared via common ordering metrics: Hamming distance and Kendall‘s Tau. The final formula is:

$$\text{LRscore} = \alpha(d \cdot bp) + (1 - \alpha)\text{BLEU}, \tag{2.26}$$

where d is the distance between the two permutations (Hamming or Kendall, or their weighted mean). *bp* is a brevity penalty. Both d and *bp* are estimated for each sentence separately and then averaged over the test set.

We use the implementation provided by the authors of the LRscore.[11] The implementation does not come come with pre-tuned hyper-parameters, and besides the paper shows that different settings work better for different language pairs (Birch and Osborne, 2010). Having no conclusive argument in favour of a specific hyper-parameter setting, we set d to the uniform average of the two distance metrics, and α is set to 1, thus giving BLEU 0 weight, as we use it separately anyway in our experiments. In order to compute alignments for the test sets, which are needed to compute the score, we concatenated the parallel text with an additional 250K lines of parallel text from the training data (see Section 3.5.3) to ensure better generalization of the alignment algorithm (implemented in GIZA++ (Och and Ney, 2003a)).

2.3.2 Statistical significance testing

Approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005) is used to detect statistically significant differences between outputs of different systems. Riezler and Maxwell (2005) have shown that approximate randomization is less sensitive to Type-I errors than bootstrap resampling (Koehn, 2004) in the context of machine translation.

We use our in-house implementation to run approximate randomization tests for BLEU and TER and MultEval (Clark et al., 2014)[12] for METEOR.

[10] TER COMpute Java code, version 0.7.25, `http://www.cs.umd.edu/~snover/tercom/`.

[11] `http://homepages.inf.ed.ac.uk/abmayne/code/lrscore.tar.gz`

[12] `https://github.com/jhclark/multeval`

Part I

Syntax-based Bilingual Language Models for Statistical Machine Translation

3 Background: Concepts, Related Work, Baseline

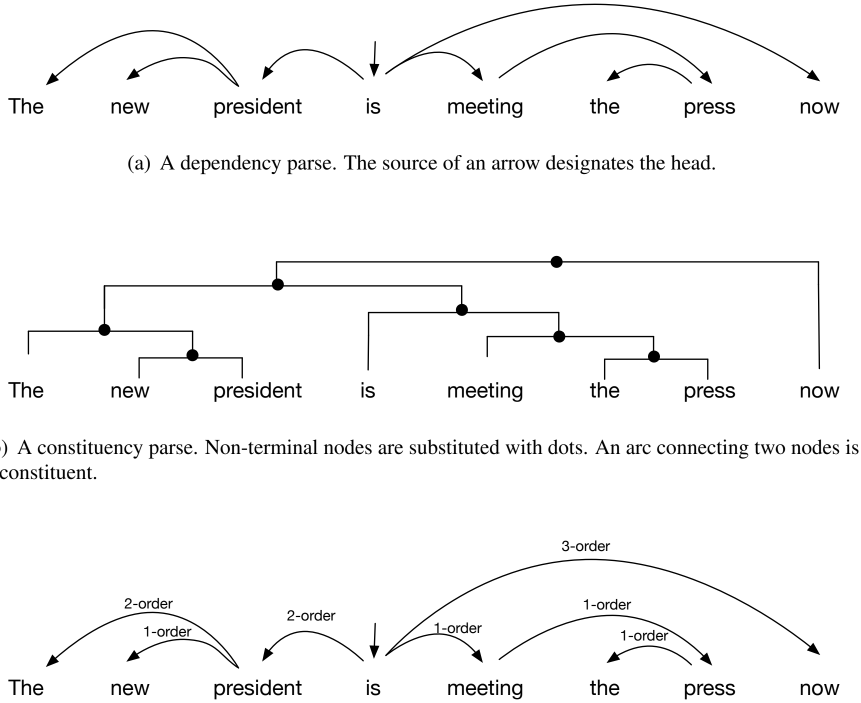
The purpose of this chapter is to provide the background relevant to Chapters 4 and 5 of Part I of the thesis. Namely, both chapters propose syntax-based language models, therefore we introduce the basics of the corresponding syntactic formalisms (Section 3.1) and provide an overview of syntax-based methods in SMT from the literature (Section 3.2). Additionally, we give some background on the bilingual language models (BiLMs (Niehues et al., 2011)) which we extend with syntactic representations in Chapter 4 (Section 3.3), and introduce the structured language models (SLMs (Chelba and Jelinek, 2000)) which we integrate into phrase-based SMT in Chapter 5 (Section 3.4). Finally, we provide a specification of the data and training setup of the phrase-based SMT system used in the experiments of Chapters 4 and 5 (Section 3.5).

3.1 Constituency and dependency formalisms

One of the fundamental propositions of linguistics is that sentences in a natural language have a latent structure that is more complex than the observed linear sequence of words. The study of this latent structure is called *syntax*. A *syntactic formalism* is a formal language that describes the latent structure of sentences. A *formal grammar* specifies the mapping between natural language sentences and their representations in a specific formalism. Good syntactic formalisms and grammars are the ones that accept well-formed sentences in a language and prohibit non-well-formed sentences.

The study of syntactic formalisms is an active research field in linguistics and many formalisms have been proposed (Chomsky, 2002; Lucien, 1959; Moot and Retoré, 2012; Bresnan et al., 2001). However, the most common formalisms used in natural language processing are the most basic ones: the constituency formalism (Chomsky, 2002) and the dependency formalism (Lucien, 1959). Both of them are based on the idea that sentences have a *hierarchical* latent structure. Constituents and dependencies formalize these hierarchical relations in different ways.

We start with an example: *The president is meeting the press.* It can be argued that the form of the word *is* in this sentence is determined by the fact that it is preceded by the word *president* (a phenomenon called agreement). From the point of view of meaning, we can say that this sentence expresses an event of a meeting between one



(a) A dependency parse. The source of an arrow designates the head.

(b) A constituency parse. Non-terminal nodes are substituted with dots. An arc connecting two nodes is a constituent.

(c) A labeled dependency parse representing the constituency parse from Figure 3.1(b).

Figure 3.1: Examples of parses in different syntactic formalisms. The example does not necessarily represent a ‘correct’ annotation scheme, but is only intended to illustrate the idea behind the formalism.

specific entity *president* and another specific entity *press*. We can expand the sentence to *The new president is meeting the press now*, by adding two more words. First of all, the agreement between *president* and *is* does not get affected by the insertion of *new*. Intuitively, this word just modifies *president*, but does not affect its relations with other words. The addition of the two words also does not change the core semantics of the sentence. It just adds a few details (how long the president has had his status and when exactly the meeting is happening), but does not change what the ‘more important’ words already express. Linguistics provides us with a lot more arguments supporting the view that words and larger units in a sentence are related to each other hierarchically.

The dependency formalism sees a sentence as a directed acyclic graph, where the words are the nodes. A directed edge, also referred to as arc, between two words represents an asymmetric dependency relation. A dominant word is said to be the *head* and the dominated word is the *modifier*. In the example above, *president* is the head and *new* is the modifier. The hierarchical relations in a sentence discussed in the previous paragraph derive from this basic dependency relation. We should point out that there are multiple conventions in the field of how to graphically represent a dependency

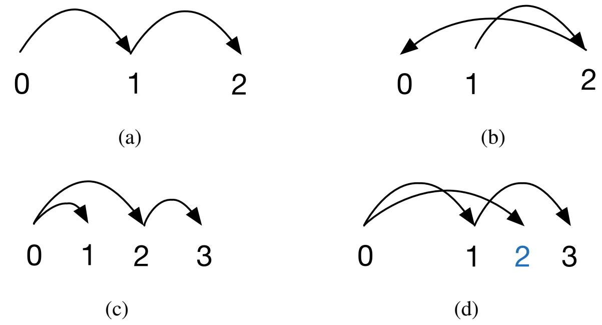


Figure 3.2: Examples of projective and non-projective parses. (a-b): projective (a) and non-projective (b) parses corresponding to isomorphic dependency trees. (b) is non-projective because node 1 is not a descendant of either 0 or 2 (it is the parent of 2). (c-d): projective (c) and non-projective (d) parses corresponding to isomorphic dependency trees. Node 2 in (d) is placed between its sibling (node 1) and the child of its sibling (node 3), neither of which is its ancestor.

arc. Some researchers interpret the source of the directed edge as the head and the target as modifier, and some do the opposite. Here, we adopt the former convention. A dependency-parsed example can be found in Figure 3.1(a).

The constituency formalism groups continuous strings of words[1] in a sentence into constituents, which behave like a unit, i.e., they can be meaningfully substituted by another constituent or can be used as a building block in another sentence. In each constituent one element is marked as dominant. A constituent can be an element in a larger constituent. For instance, in the example above, *president* is a singleton constituent, which is included in a constituent *new president*. A constituency structure can be represented as an undirected tree with terminal nodes (words) and non-terminal nodes (labels of constituents). Figure 3.1(b) provides an example of a constituency-parsed sentence.

The two formal definitions impose different constraints on the class of possible hierarchical structures. Constituency is a stricter formalism, since it requires two units related to each other by an immediate constituency relation to be adjacent in a sentence. Dependency relations, on the other hand, do not change with the linear order of words and therefore can provide a characterization of a word's syntactic class that is invariant under word ordering. In fact, every constituency tree can be formulated as a dependency tree with additional labeling (designating how deeply a word is nested in a constituency structure), but not vice versa (see example in Figure 3.1(c)). However, in practice, a projectivity constraint is assumed, which prohibits linear orders inconsistent with the hierarchical structure. Formally, if we denote a sentence as W and its dependency structure as D, where $D(w_i, w_j)$ means that w_i is the head and w_j the modifier, then D is a *projective* structure if: For every word pair $w_i, w_j \in W$ so that $D(w_i, w_j)$

[1] In the common version of the formalism, a constient is a group of two adjacent words, thus being a binary constituency formalism.

it holds that every $w_k \in W$ so that $i < k < j$ or $j < k < i$ is a descendant of w_i, i.e., $D^*(w_i, w_k)$.[2] Figure 3.2 provides examples of projective and non-projective dependency structures.

3.2 Syntax in statistical machine translation

In this section we provide an overview of methods grounded in syntactic formalisms used in machine translation. It is by no means an exhaustive overview. Our goal is to outline *types of approaches*, with a specific focus on methods relevant to Chapters 4 and 5.

As we explained in Section 3.1, syntactic formalisms express hierarchical relations between the elements of a sentence. These hierarchical relations comprise a part of what is expressed by the sentence. Therefore, it is relevant to machine translation which aims to translate what is expressed by a sentence in one language into another language. In Section 5.2 of the chapter on strunctured languaged models applied to SMT, we discuss the underpinnings of this idea in more detail. This idea can be exploited in machine translation in a number of ways, as discussed below.

First of all, syntactic correspondence can be used to constrain or even derive translation correspondence. Eisner (2003) start with a parallel unaligned corpus of parsed sentences and derive sub-sentential translation correspondents from it. Lavie et al. (2008) start from a word-aligned parsed parallel corpus and then iteratively refine translation correspondence on a sub-sentential level. Different syntax-based SMT models (Yamada and Knight, 2001; Liu et al., 2006; Huang et al., 2006; Marton and Resnik, 2008; Shen et al., 2008) extract translation rules only if they correspond to well-formed syntactic structures and comply to word alignments.

Instead of having syntax determine the translational units of the model, one can integrate syntax as an additional feature function in the log-linear scoring function, see Equation 2.3 in Section 2.1, to constrain the translation correspondence at decoding time. For example, Ge (2010) defines binary features characterizing what parts of a source parse tree have been translated or are currently being translated. Cherry (2008); Bach et al. (2009) formulate a cohesive constraint that fires when some subtree of a source parse tree has been partially but not fully translated and the current source words to be translated are not inside this subtree, see also Section 5.2 for more details on this model. Some syntax-based features are probabilistic models in their own right, modeling a specific aspect of translation such as reordering (Chang and Toutanova, 2007; Lerner and Petrov, 2013). Our approach in Chapter 5 falls into this category.

Another line of research in SMT directly relates translation correspondence and hierarchical structure of sentences to each other, but does not rely on external language-specific parses at all. Instead, this kind of approach derives the structure from the word alignments and makes it part of the model (Wu, 1997b; Chiang, 2007; Stanojevic, 2015).

Finally, since syntactic structure comprises part of what a sentence expresses, it can be used as a way to provide richer representations of predefined translation units (Zollmann and Vogel, 2011; Li et al., 2012; de Buy Wenniger and Sima'an, 2013). Our contribution in Chapter 4 falls into this category.

[2] $D^*(w, v)$ if $D(w, v)$ or if $\exists u$ so that $D(w, u) \wedge D^*(u, v)$.

Motivated by the same ideas about the relation between the structure of a sentence and the translation correspondence, there has also been research on integrating syntax in neural machine translation. So far, most of the methods propose a way to encode source syntactic structure in order to obtain a more informative representation of the input (Eriguchi et al., 2016; Bastings et al., 2017).

3.3 Bilingual language models

The phrase *bilingual language model* can in principle denote any language model involving elements from both the source and the target sentence, including, for instance, the bilingual structured language models from Chapter 5. However, in this thesis we use it as a term for a specific class of language models, namely, the models used in Chapter 4. Specifically, a bilingual language model (BiLM) is an n-gram model, see the definition in Equation 2.9, with elements consisting of positions from the source and target sentences related to each other by word alignment.

We distinguish between *bilingual tokens* and their representations. A bilingual token is a tuple of source and target sentence positions related by word alignment in a particular way. A representation of a bilingual token is a way in which one chooses to represent the positions inside a token. A definition of a BiLM thus consists of a choice of segmentation (into tokens) and representation, given a source and target sentence and a word alignment between them.

A number of segmentation algorithms have been proposed in the literature, most of which are guided by a subset of the following properties: exhaustive, monotonic, and minimal. By exhaustive segmentation we understand a segmentation that produces tokens such that for every position inside of it all of its aligned positions are inside the same token. Monotonic segmentation is such that in a resulting sequence of tokens $t_1 ... t_n$, for every consecutive t_i, t_{i+1}, every source position[3] $f' \in t_i$ linearly precedes in F every source position $f'' \in t_{i+1}$, and likewise for target positions. Minimal segmentation produces a sequence of tokens none of which can be further decomposed into well-formed tokens. Requiring a segmentation to be monotonic or exhaustive increases the size of the resulting vocabulary of tokens, thus potentially causing model sparseness, but may capture important data patterns. Requiring a segmentation to be minimal eliminates the ambiguity of segmentation at the inference stage. Segmentation into phrases, see Section 2.1.2, is exhaustive, but not monotonic and not minimal. Marino et al. (2006); Durrani et al. (2011) propose segmentations that satisfy all of the three properties. Crego et al. (2005); Niehues et al. (2011) propose segmentations that are minimal, exhaustive, but not monotonic. In Chapter 4 we work with a BiLM segmentation proposed in (Niehues et al., 2011), see Equation 4.1 for the full formal definition.

Various representations have been proposed in the literature as well. While the choice of a segmentation determines what one chooses as a basic operational unit when modeling source-target correspondence, a representation determines on what aspect of the correspondence the model focuses. The default choice of representation is lexical:

[3] We use the same notation for f for a source word and a source position.

using surface word forms corresponding to the positions. However, lexical representation may result in an extremely large token vocabulary, especially for languages with large word vocabularies, thus failing to generalize well. A common choice of a more abstract representation is syntax-based annotation, such as part of speech (Niehues et al., 2011), class-based annotation (Durrani et al., 2013), rich syntactic and morphological annotation (Crego and Mariño, 2006; Crego and Habash, 2008). In Chapter 4 we propose representations derived from source dependency parses.

3.4 Structured language models

N-gram language models (Section 2.1.4) are the most common type of language models used in natural language processing and specifically machine translation. However, their major shortcoming is that by design they model relations between elements of a sequence at a short distance by being restricted to the span of n elements. In Section 3.1 (on syntactic formalisms for natural language) we introduced the idea that words and larger elements in a sentence are related hierarchically. Consequently, some words or phrases in a sentence are high in the hierarchy derived from the structure of a sentence and it is important to model their mutual relations correctly. Moreover, the modifiers of these dominant words can be arbitrarily large which can result in the actual distance between the dominant words to be too large to be captured by an n-gram model. This phenomenon is commonly referred to as long-distance dependencies. Consider the sentences *“A man walked in”* and *“A man who bought a house walked in”*): the words *man* and *walked* arguably represent the core of what the sentence expresses. However, an n-gram model with insufficiently large order would not be able to condition *walked* on *man*. If the order of the model was large, then it would be hard to reliably estimate it.

Structured language models (SLMs (Chelba and Jelinek, 2000; Charniak, 2001; Roark, 2001; Pauls and Klein, 2012; Gubbins and Vlachos, 2013)) are designed to address this shortcoming of n-gram models. The linguistic intuition behind SLMs is that the modifiers of a word do not essentially change its distributional properties but just provide additional specification. In Figure 3.3(a) the word *president* has two modifiers: *the* and *former* and it follows *yesterday* and precedes *met*. If instead its modifier was *a* or an entire relative clause, the placement and the semantics (in the context of the sentence) *yesterday* and *met* stay the same. To capture this observation, SLMs model the generation of a sentence as simultaneously building up of a sequence of words and the hierarchical structure that characterizes them.

In this thesis we concentrate on incremental (left-to-right) SLMs (Chelba and Jelinek, 2000; Charniak, 2001; Gubbins and Vlachos, 2013). In Chapter 5, we work with an adaption of the model from (Chelba and Jelinek, 2000) that we describe here. Most other incremental SLMs follow the same logic. Chelba and Jelinek (2000) model an incremental (left-to-right) generation of a parsed sentence as a sequential prediction of a word w_i conditioned on the partial sequence W_{i-i} and its corresponding parse Tree^{W}_{i-1}, followed by a prediction step resulting in extending the partial parse to Tree^{W}_{i}. Chelba and Jelinek (2000) use constituency parses where the terminal nodes (i.e., the surface words) are directly dominated by nodes with the corresponding part of speech (POS) tags. Therefore, they decompose the step of producing Tree^{W}_{i} conditioned on Tree^{W}_{i-1}

and w_i into first predicting the POS tag t_i of w_i and then deciding how to incorporate (w_i, t_i) into Tree^W_{i-1}:

$$
\begin{aligned}
p_{\mathit{SLM}}(W, \mathit{Tree}^W) = \prod_{i=1}^{|W|} & p(w_i | W_{i-1}, \mathit{Tree}^W_{i-1}) \\
& \cdot p(t_i | w_i, W_{i-1}, \mathit{Tree}^W_{i-1}) \\
& \cdot p(\mathit{Tree}^W_i | w_i, t_i, W_{i-1}, \mathit{Tree}^W_{i-1})
\end{aligned}
\tag{3.1}
$$

We first briefly describe the parsing model $p(\mathit{Tree}^W_i | w_i, t_i, W_{i-1}, \mathit{Tree}^W_{i-1})$. This will help us to explain how p_{SLM} manages to capture the structural relations between words that we talked about at the beginning of the section, without running into sparsity issues due to long-distance dependencies.

In order to model $p(\mathit{Tree}^W_i | w_i, p_i, W_{i-1}, \mathit{Tree}^W_{i-1})$, Chelba and Jelinek (2000) use a shift-reduce parsing algorithm. At a high level, a shift-reduce parsing procedure consists in scanning a sequence from left to right, while having an auxiliary stack structure that keeps track of the partial parse (constructed up to the current step). Each next element w_i in a sequence is added to a stack as a singleton tree (an isolated node), a parsing operation called *shift*. After this the parser predicts a sequence of *reduce* operations until it predicts the *null* action, after which the next element in a sequence is scanned. A *reduce-left* operation combines the topmost element e_i in a stack with the subsequent element e_{i-1} into a common structure (a constituent or a dependency arc), so that the e_i is the dominant element. *Reduce-right* does the same operation, but with the element e_{i-1} becoming the dominant element of the resulting structure. For more details on the parsing procedure, refer to (Aho et al., 1986; Chelba and Jelinek, 2000). This formal procedure entails that each time a word is scanned, a stack contains a sequence of disjoint tree structures each of which has a dominant element. The term Tree^W_{i-1} above denotes this sequence of subtrees.

The definition of the parsing procedure directly entails how the hierarchical relations between words are captured by SLMs. Given a subsequence W_{i-1} and its associated parse Tree^W_{i-1}, the roots of all the disconnected subtrees in Tree^W_{i-1} are called *exposed heads* (Chelba and Jelinek, 2000). Consider Figure 3.3(a) again. In a left-to-right scenario, when *met* is generated, a regular n-gram LM conditions it on *yesterday the former president*, while an SLM conditions it on *yesterday president*, since these two words are the exposed heads with respect to *met* (Figure 3.3(b)). The words *the* and *former* are modifiers of *president* and they are filtered out. Thus we obtain a less specific conditioning history, which may lead to the resulting model being less sparse. Another benefit is that SLMs can capture long-distance relations. If *president* had as its modifier a relative clause (Figure 3.3(c)) then a simple n-gram LM would be conditioned on *days before* (assuming $n = 3$), while an SLM would condition *met* on *yesterday president*.

Similar to an n-gram model, one can restrict the conditioning history to the topmost $n - 1$ tree roots on a stack. Thus, the word prediction model can be expressed as:

$$
p(w_i | W_{i-1}, \mathit{Tree}^W_{i-1}) = \mathit{Expos}(W_{i-1}, \mathit{Tree}^W_{i-1}), \tag{3.2}
$$

where $\mathit{Expos}(W_{i-1}, \mathit{Tree}^W_{i-1})$ are the words corresponding to the exposed heads on the stack (or n topmost exposed heads).

The POS tag prediction model can be expressed likewise, where we designate the POS labels of the exposed heads on the stack as $\mathit{ExposPOS}(W_{i-1}, \mathit{Tree}^W_{i-1})$:

$$p(t_i|w_i, W_{i-1}, \mathit{Tree}^W_{i-1}) = \mathit{ExposPOS}(W_{i-1}, \mathit{Tree}^W_{i-1}) \tag{3.3}$$

In Chapter 5 we work with a dependency variant of Chelba and Jelinek (2000)'s SLM (similar to (Gubbins and Vlachos, 2013)). We omit the step of generating the POS tags of words and thus the overall formula becomes:

$$\begin{aligned} p_{SLM}(W, \mathit{Tree}^W) = \prod_{i=1}^{|W|} & p(w_i|\mathit{Expos}(W_{i-1}, \mathit{Tree}^W_{i-1})) \\ & \cdot p(\mathit{Tree}^W_i|w_i, \mathit{Expos}(W_{i-1}, \mathit{Tree}^W_{i-1})), \end{aligned} \tag{3.4}$$

3.5 PBSMT baseline and experimental setup

In this section we provide a specification of the phrase-based SMT baseline system used in our experiments in Chapters 4 and 5. Here we only specify the choice of models, data, and hyperparameters. For more background on phrase-based SMT, consult Section 2.1 and the references provided there.

We specify the choice and statistics of the training and testing data sets in Section 3.5.1, the data preprocessing steps in Section 3.5.2, and the details of the training and decoding algorithm in Section 3.5.3.

3.5.1 Data

In both Chapters 4 and 5 we perform experiments on the Arabic-English and Chinese-English language pairs. We use the standard publicly available training data and testing benchmarks.

The following Arabic-English corpora are used: LDC2006E25, LDC2004T18, LDC2004T17, LDC2005E46, LDC2007T08, LDC2004E13. The following Chinese-English parallel corpora were used: LDC2002E18, LDC2002L27, LDC2003E07, LDC2003E14, LDC2005T06, LDC2005T10, LDC2005T34, and several Gale corpora. Statistics about the preprocessed training dataset for both language pairs are summarized in Tables 3.1 and 3.2 for Arabic-English and Chinese-English, respectively. Since both Chapters 4 and 5 involve models where source-side parses are used, we also provide statistics about the parsed subset (see Section 3.5.2 for information about the parsers).

For testing, we use the test benchmarks provided by OpenMT[4]. We use the following test sets for Arabic-English: MT02, MT03, MT05, MT06nist[5], MT08 and MT09. We used the following for Arabic, and MT02, MT03, MT05, MT06nist[5] and MT08 for

[4] `https://catalog.ldc.upenn.edu/LDC2013T07`

[5] We refer to it as MT06. in the next two chapters.

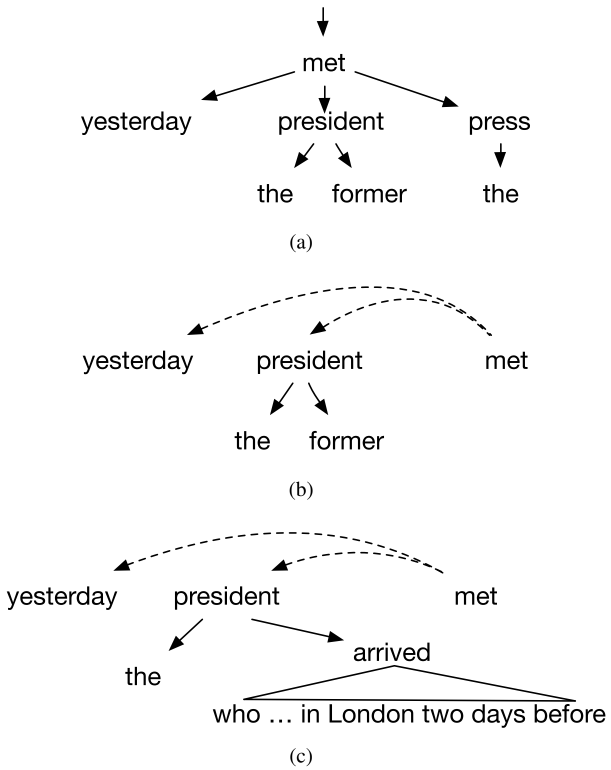


Figure 3.3: A fully parsed sentence (a) and its partial parse (b) during sequential generation. The partial parse in (b) has two disconnected subtrees with roots *yesterday* and *president*. These roots are the exposed heads for *met*. (c) is an alternative sentence with a similar structure: *president* is still a root of a subtree, and thus and an exposed head.

Chinese. For feature weight tuning, see Section 2.1.7, we use OpenMT's MT04 for both language pairs.

The target-side n-gram language model, see Section 2.1.4, is trained on the English Gigaword corpus (LDC2003T05).

3.5.2 Data preprocessing and labeling

The general preprocessing steps are, in the following order: tokenization, lowercasing, and deduplication. For both Arabic and Chinese, we use the Stanford CoreNLP tokenizers (Monroe et al., 2014; Tseng et al., 2005) with the Penn Arabic Treebank tokenization standard and the Chinese Penn Treebank tokenization standard. For English we use a simple in-house tokenizer.

Our models in Chapters 4 and 5 involve dependency grammar[6] analyses of the

[6] See Section 3.1 for the basics of the dependency formalism.

Table 3.1: Training data for Arabic-English experiments in Part I of the thesis. In our experiments we use the parses of the source sentence, therefore only the parsed subset of the source training set can be used. The parsed subset is smaller than the full training set because the parser failed to compute a well-formed parse for some sentences. The type/token statistics is computed after data preprocessing (Section 3.5.2).

Training set	**N. of lines**	**N. of word tokens**		**N. of word types**	
		source	target	source	target
full training set	4,376,320	148M	146.1M	0.5M	0.3M
source-parsed subset	941,171	29M	29.8M	0.2M	0.2M

Table 3.2: Training data for the Chinese-English experiments in Part I of the thesis. In our experiments we use the parses of the source sentence, therefore only the parsed subset of the source training set can be used. The parsed subset is smaller than the full training set because the parser failed to compute a well-formed parse for some sentences. The type/token statistics is computed after data preprocessing (Section 3.5.2).

Training set	**N. of lines**	**N. of word tokens**		**N. of word types**	
		source	target	source	target
full training set	2,104,652	20.2M	28.2M	1.7M	0.9M
source-parsed subset	867,861	18.2M	26.1M	1.1M	0.13M

source side, and part-of-speech analyses (POS) of the target side. For parsing of the Arabic side, we use a constituency parser from the Stanford CoreNLP package (Green and Manning, 2010), since a dependency parser was not available. We extract the dependency structures from the computed constituency structures based on the rules in (Collins, 1999). For Chinese, we use the Stanford dependency parser (Chang et al., 2009). For POS-tagging of the English target side, we use the POS-tagger from the Stanford CoreNLP package (Toutanova et al., 2003).

3.5.3 Model training and testing

A structured description of a PBSMT pipeline that we use as a baseline is provided in Section 2.1.3. Here we only report the hyper-parameter settings and the choice of software.

Word alignments, see Section 2.1.1, are computed with GIZA++ (Och and Ney, 2003a). Phrase pairs are extracted with maximum length of 7 on both source and target sides. The translation models (see Section 2.1.3) are estimated with relative counts using the Moses[7] phrase-table building script (Koehn et al., 2007). A 5-gram target

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

language model, see Section 2.1.4, is trained using SRILM[8] (Stolcke et al., 2011) with modified Kneser-Ney smoothing and interpolation (Chen and Goodman, 1996). We use an in-house implementation by Christof Monz of the lexicalized distortion models, see Section 2.1.5.

For tuning and decoding we use in-house implementation by Christof Monz of a PBSMT system similar to Moses (Koehn et al., 2007). We use the following decoding settings (see Section 2.1.6): the distortion limit is set to 5, the stack size is 100, the beam width is 0.1, and the maximum number of expansions is 30 per partial hypothesis in a stack.

The feature weights are tuned by using pairwise ranking optimization (PRO (Hopkins and May, 2011)) with an in-house implementation (see Section 2.1.7). During tuning, 14 PRO parameter estimation runs are performed in parallel on different samples of the n-best list after each decoder iteration. The weights of the individual PRO runs are then averaged and passed on to the next decoding iteration. Performing weight estimation independently for a number of samples corrects for some of the instability that can be caused by individual samples.

[8] `http://www.speech.sri.com/projects/srilm/`

4 Dependency-Based Bilingual Language Models for Reordering in Statistical Machine Translation

4.1 Introduction

In this chapter we focus on models based on the syntactic representations of parallel sentences and designed to improve reordering. In Chapter 3 we reviewed some previous approaches to reordering in SMT based on syntactic representation of sentences. We focused on approaches that model the generation of translation and have reordering as a side-effect. This class of models can be contrasted to methods that either focus on mutual order of words or phrases or characterize some aspects of reordering (such as length of a jump). Namely, we reviewed two approaches that we combine in this work. The first class of approaches is bilingual language models (BiLMs) (Marino et al., 2006; Niehues et al., 2011). Instead of directly characterizing reordering, they model sequences of elementary translation events as a Markov process. The original works on BiLMs (Marino et al., 2006; Niehues et al., 2011) have mostly used lexical information to represent elementary bilingual tokens, although Crego and Yvon (2010a) and Crego and Yvon (2010b) label bilingual tokens with a rich set of POS tags. A second class of approaches we reviewed was various syntactic methods whereby reordering is characterized in terms of restructuring of the source syntactic parse tree. Tree-based approaches in SMT incorporate syntactic information in the representation of the source sentence (Liu et al., 2006; Huang et al., 2006; Marton and Resnik, 2008), target sentence (Shen et al., 2008), or both (Chiang, 2007, 2010). Such a representation allows one to have a more detailed definition of translation events and to redefine decoding as parsing. Reordering is thus a result of a given parse derivation. A top-down derivation captures better the *global* structure of the sentence than a simple PBSMT decoding algorithm, and therefore is more likely to provide a more accurate model of translation. At the same time, parsing-based approaches are a lot more complex and require more intricate optimization and estimation techniques, see (Huang and Mi, 2010).

The idea behind this chapter is to explore the trade-off between global syntax-aware modeling of a translation process and simpler models with fixed sized context. We would like to keep the simplicity of PBSMT but move towards the expressiveness typical of tree-based models. We propose to incrementally build up the syntactic representation of a translation during decoding by adding precomputed fragments from the source parse tree. We mostly use source syntactic information to characterize reordering, since during decoding we have access to the entire source sentence. This allows us to obtain a better syntactic analysis for it (than for a partial sentence) and to precompute the units that our model operates with. This idea to combine the merits of the two SMT paradigms has been proposed before, where Huang and Mi (2010) introduce incremental decoding for a tree-based model. On a general level, our approach is similar to theirs in that it keeps track of a sequence of source syntactic subtrees that are being translated at consecutive decoding steps. An important difference is that they keep track of whether the visited subtrees have been fully translated, while in our approach, once a syntactic structural unit has been added to the history, it is not updated anymore.

This brings us back to **RQ1** of this thesis, which we repeat in its entirety here:

RQ1 Can we improve reordering by modeling sequences of syntactic structures representing basic operational units of translation?

RQ1.a. Can the representations only include the local syntactic information of a node in a syntactic parse? What is the minimum context that the local representation should incorporate?

RQ1.b. How do local syntactic representations compare to representations including explicit lexical information of the basic translational units?

RQ1.c. What kind of reordering phenomena are captured by such models?

In this chapter we focus on the "lower bound" aspect of this research question: we aim to show that local syntactic representations are better or complementary as compared to local lexical information (i.e., representing tokens as words). We do not experimentally compare our proposed model to a more sophisticated model of syntactic restructuring.

The contributions of this work can be summarized as follows:

1. We adopt the definition of a bilingual token from BiLMs as proposed by Niehues et al. (2011) (Section 3.3) and propose a novel token representation to better capture the reordering process (Section 4.2). We represent bilingual tokens as *local* syntactic contexts of the source and target positions included in a bilingual token (Section 4.3).

2. We investigate different degrees of syntactic locality that are necessary and/or sufficient to capture the reordering process (Section 4.3).

3. We experimentally evaluate the methods on language pairs characterized by complex reordering patters: Chinese-English (Wang et al., 2007), Arabic-English (Elming and Habash, 2009; Carpuat et al., 2010) (Section 4.4).

4. We evaluate our models against the baseline[1] and two systems consisting of a baseline and a BiLM from the literature. The purpose of a comparison to the

[1] See Section 3.5.3 for the baseline specifications.

former is to validate the necessity for complex contextual representations (and thus partially answer **RQ1.a**), and the purpose of the latter is to answer **RQ1.b**.

5. We evaluate the effect of the models on MT performance with general-purpose metrics. Additionally, we analyze the effect on reordering specifically, by looking into reordering-sensitive metrics and by decoding in an extended search space that allows for more long distance reordering (Section 4.4).

4.2 Choosing a BiLM to model reordering

In the background chapter (Section 3.3) we distinguish between the notions of a bilingual token and its representation. A bilingual token is defined as a tuple of source and target positions constrained by the given sentence-internal word alignment. One can choose different ways to represent the positions inside a token. A typical representation is lexical, however, part of speech based representations and representations based on detailed morphological annotation have also been proposed (Crego and Yvon, 2010b; Niehues et al., 2011). In this section we motivate our bilingual token definition of choice (based on Niehues et al. (2011)), and discuss the pros and cons of the different representations in the context of modeling reordering in MT. The main claim of this section is that lexical and simple syntactic representations are empirically often not expressive enough to differentiate between alternative reorderings. This leads to the subsequent section, where we devise a syntactic representation devoid of the disadvantages discussed here.

We should first note that the most commonly used n-gram model to distinguish between reorderings is a target language model, which does not take translation correspondence into account and just models target-side fluency. Al-Onaizan and Papineni (2006) experimentally show that target language models by themselves are not sufficient to correctly characterize reordering.

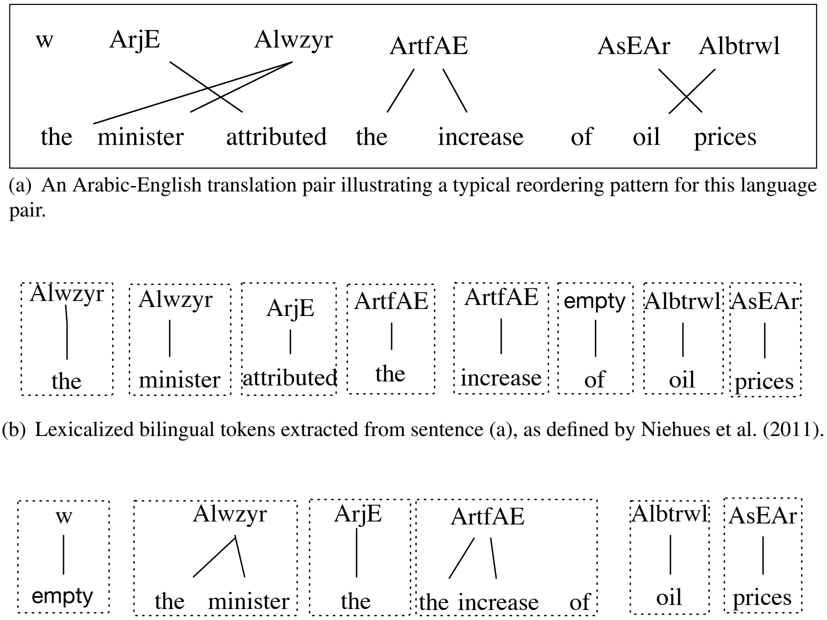
We will complement our argument below with an example of a word-aligned sentence pair in Figure 4.1.a.[2] It demonstrates a common Arabic-English reordering, whereby the main verb is sentence-initial, followed by the subject and then the object (the so-called VSO order, see Carpuat et al. (2010)).

4.2.1 Choosing the definition of a bilingual token

We choose to use the definition of a bilingual token from (Niehues et al., 2011). Given the source sentence $F = \langle f_1, ..., f_n \rangle$, target sentence $E = \langle e_1, ..., e_m \rangle$, and the word alignment between them which we choose to formalize as a mapping from the target words to the powerset of source words $\mathcal{A} : E \to \mathcal{P}(F)$, we can extract a sequence of bilingual tokens $\langle t_1, ..., t_m \rangle$ as follows:

$$t_i = \langle e_i, \{f | f \in \mathcal{A}(e_i)\} \rangle, \tag{4.1}$$

[2] We used Buckwalter transliteration for Arabic words. We thank Arianna Bisazza for help with the transliteration.



(a) An Arabic-English translation pair illustrating a typical reordering pattern for this language pair.

(b) Lexicalized bilingual tokens extracted from sentence (a), as defined by Niehues et al. (2011).

(c) Lexicalized bilingual tokens extracted from sentence (a), as defined by Durrani et al. (2011) (MTUs).

Figure 4.1: Arabic-English parallel sentence, automatically word-aligned. The bilingual token sequences are produced according to two alternative definitions of a bilingual token.

With this definition we would decompose our running example in Figure 4.1.a into a sequence in Figure 4.1.b (we use lexicalized representation of tokens in this example, but it need not be).

In our choice of a segmentation definition we are guided by the fact that with this definition we can unambinguously extract a sequence of bilingual tokens. This implies that there is no hidden segmentation variable and thus ensures simpler learning and inference. On the contrary, a definition of a phrase (as in phrase-based translation) is an example of an ambiguous segmentation. The non-ambiguity of our BiLM model of choice has two attractive consequences. The first one is minimal vocabulary size (given a fixed representation) as compared to an ambiguous segmentation model. The second one is unambiguous representation of reordering. For example, two different segmentations of ba into $[ba]$ and $[b][a]$ still represent the same permutation of the sequence ab.

Another popular method for unambiguous bilingual segmentation into tokens would be the minimal translation units (MTUs) by Durrani et al. (2011). Figure 4.1.c demonstrates the tokenization extracted with the MTU defintion. Since Niehues et al. (2011) have shown their model to work successfully as an additional feature in combination with commonly used standard phrase-based features, we use their approach as the main point of reference and base our approach on their segmentation method. In the rest of the chapter, we use *bilingual token* to talk about the tokens from (Niehues et al., 2011). At the same time, we do not see any specific obstacles for combining our work with MTUs.

4.2.2 Suitability of lexicalized BiLM representation to model reordering

As proposed in the introduction, lexical information is not very well-suited to capture reordering regularities. Consider Figure 4.2.a. The extracted sequence of bilingual tokens is produced by aligning source words with respect to target words (so that they are in the same order), as demonstrated by the shaded part of the picture. In Figure 4.2.a, if we substituted the Arabic translation of *Egyptian* for the Arabic translation of *Israeli* in the fourth token, the reordering should remain the same. What matters for reordering is the syntactic role or context of a word. By using unnecessarily fine-grained categories we risk running into sparsity issues.

Niehues et al. (2011) also described an alternative variant of the original BiLM, where words are substituted by their POS tags (Figure 4.2.a, shaded part). Also, however, POS information by itself may be insufficiently expressive to separate correct and incorrect reorderings, see Figure 4.2.b. Although the corresponding sequence of POS-tag-substituted bilingual tokens is different from the correct sequence (Figure 4.2.b, shaded part), it still is a likely sequence. To illustrate this point, we computed the log-probabilities of the two sequences with respect to a 4-gram BiLM model.[3] The result is that the incorrect reordering gets a higher probability of -10.25 for the incorrect reordering than the correct one (-10.39).

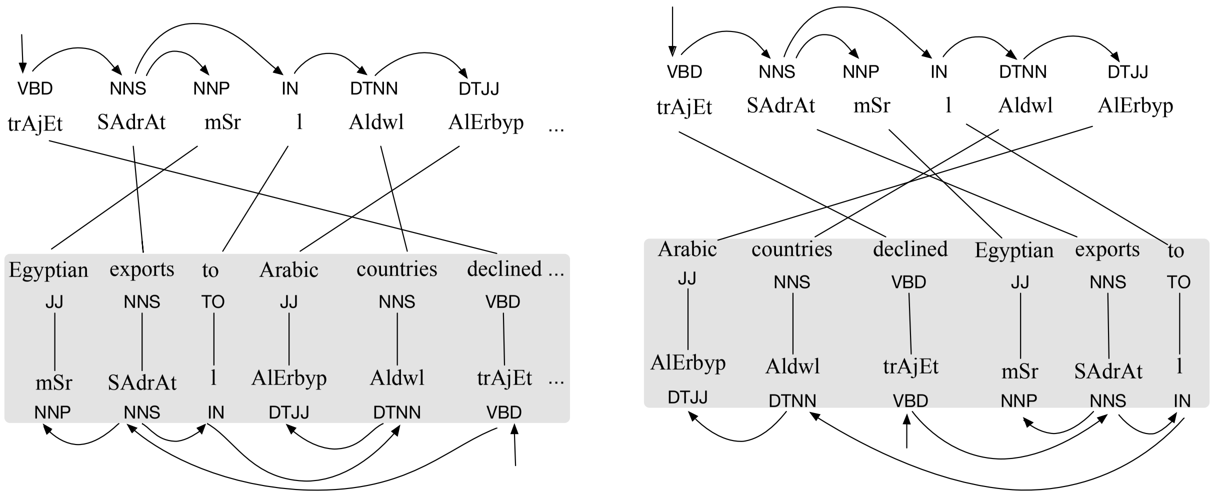
Since fully lexicalized bilingual tokens suffer from data sparsity and POS-based bilingual tokens are insufficiently expressive, the question is which level of syntactic information strikes the right balance between expressiveness and generality.

4.2.3 BiLMs with syntactic representation

Dependency grammar is commonly used in NLP to formalize role-based relations between words. The intuitive notion of syntactic modification is captured by the primitive binary relation of dependence (see Section 3.1 on details about the dependency formalism). Dependency relations do not change with the linear order of words (Figure 4.2) and therefore can provide a characterization of a word's syntactic class that is invariant under word ordering in a single language and under reordering for a pair of languages. Being simpler than the constituency formalism, it is also easier to adapt to the task of token labeling.

If we incorporate dependency relations into the representation of bilingual tokens, the incorrect reordering in Figure 4.2.b will produce a highly unlikely sequence. For example, we can substitute each source word with its POS tag and its parent's POS tag (Figure 4.3). Again, we computed 4-gram log-probabilities for the corresponding sequences: the correct reordering results in a substantially higher probability of -10.58

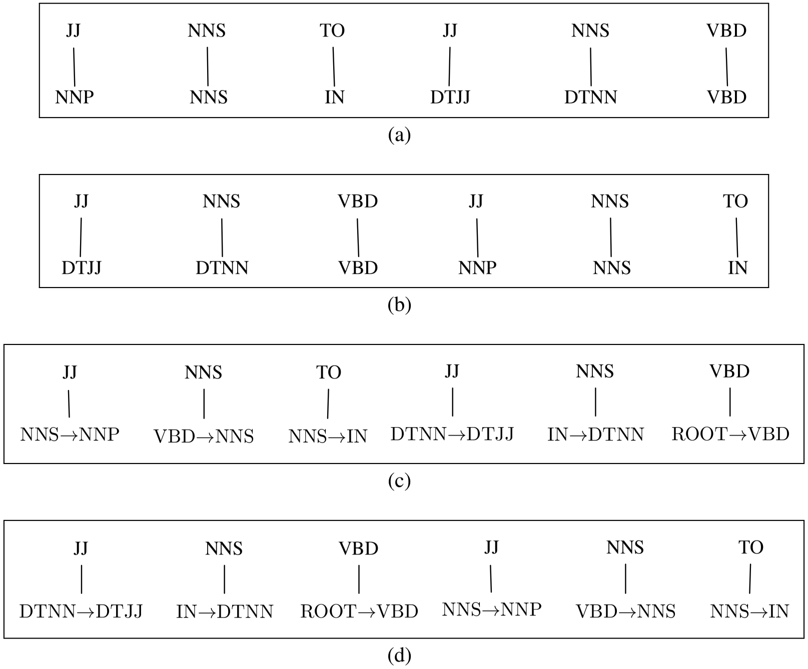
[3] Sections 3.5.3 and 4.3.3 contains details about data and software setup.



(a)

(b)

Figure 4.2: Arabic-English parallel sentence, automatically parsed and word-aligned, with corresponding sequences of bilingual tokens (in the shaded part). Comparison between translations produced via correct (a) and incorrect (b) reorderings.



(a)

(b)

(c)

(d)

Figure 4.3: Sequences of bilingual tokens with source words substituted with their and their parents' POS tags: correct (a) and incorrect (b) reorderings.

than the incorrect one (-13.48). We may consider situations where more fine-grained distinctions are required. In the next section, we explore different representations based on source dependency trees.

4.3 Dependency-based BiLM

In this section, we introduce our model which combines the BiLM from (Niehues et al., 2011) with source dependency information, *dependency-based BiLMs* (Sections 4.3.1 and 4.3.2). We refer to them as *depBiLMs* for short. We give further details on how the proposed models are trained (Section 4.3.3) and integrated into a phrase-based decoder (Section 4.3.4).

4.3.1 The general framework

In the previous section we outlined our framework as composed of two steps: First, a parallel sentence is tokenized according to the BiLM model (Niehues et al., 2011). Next, words in the bilingual tokens are substituted with their contextual representations. It is thus convenient to use the following generalized definition for a token sequence $t_1...t_n$ in our framework:

$$t_i = \langle ContE(e_i), \{ContF(f) | f \in A(e_i)\}\rangle, \tag{4.2}$$

where e_i is the i-th target word, $\mathcal{A} : E \to \mathcal{P}(F)$ is an alignment function, F and E are source and target sentences, and $ContE$ and $ContF$ are target and source *contextual functions*, respectively. A contextual function returns a word's contextual representation, based on its sentential context (source or target). See Figure 4.4 for an example of a sequence of BiLM tokens with a $ContF$ defined as returning the POS tag of the source word combined with the POS tags of its parent, grandparent and siblings, and $ContE$ defined as an identity function; see Section 4.3.2 for a detailed explanation of the functions and notation.

In this work we focus on source contextual functions ($ContF$). The main reason is the full availability of the source syntactic parse before translation. We also exploit some very simple target contextual functions, but do not go into an in-depth exploration; see (Shen et al., 2008) for an approach relying on rich target-syntactic information.

4.3.2 Dependency-based contextual functions

As discussed in the background chapter (Section 3.2), for NLP approaches exploiting dependency structure, two kinds of relations are of special importance: the parent-child relation and the sibling relation. Based on previous work, we propose to characterize contextual syntactic roles of a word in terms of POS tags of the words themselves and their relatives in a dependency tree. It is straightforward to incorporate parent information since each node has a unique parent. As for siblings information, we incorporate POS tags of the closest sibling to the left and the closest to the right. We do not include all of the siblings to avoid overfitting. In addition to these basic syntactic relations, we consider the grandparent relation.

The following list is a summary of the source contextual functions that we use. We describe a function with respect to the kind of contextual property of a word it returns:

(i) the word itself (Lex);

(ii) POS label of the word (Pos);

(iii) POS label of the word's parent; (Pos$\rightarrow$($\cdot$));

(iv) POS of a word's sibling immediately to the left, concatenated with the POS tag of the sibling immediately to the right (($\cdot$)+sibl);

(v) the POS label of the word's grandparent (Pos$\rightarrow\rightarrow$($\cdot$)).

We consider target-side contextual functions returning: (i) an empty string, (ii) POS of the word, (iii) the word itself. However, in our experiments we keep *ContF* fixed, namely: the lexicalized BiLM has the word itself as the target contextual function, and all the syntax-based functions (including depBiLMs) have POS of the target word as *ContE*.

Notation. Each of the contextual functions above by itself is not likely to be sufficient to be a good representation of a token for reordering purposes. Therefore we use combinations of these functions. We use the following notation for function combinations:

- "•" horizontally connects source (on the left) and target (on the right) contextual functions for a given model. For example, Lex•Lex refers to the original (lexicalized) BiLM.
- We use arrows ($\rightarrow$) to designate parental information (the arrow goes from parent to child). For example, Pos$\rightarrow$Pos refers to a combination of a function returning the POS of a word and the POS of its parent (as in Figure 4.3). Pos$\rightarrow$Pos$\rightarrow$Pos is a combination of the previous with the function returning the grandparent's POS.
- We use +sibl to indicate the use of the sibling function described above: For example, Pos$\rightarrow$Pos+sibl is a source function that returns the word's POS, its parent's POS and the POS labels of the closest siblings to left and right. In case there is no sibling on one of the sides, ϵ (empty word) is returned.

Finally, we use a contextual function Pos$\rightarrow$Pos$\rightarrow$Pos+sibl•Lex as an example that combines most of the primitive functions described above. Figure 4.4 represents the sentence from Figure 4.2 during decoding with the corresponding depBiLM integrated into the scoring function. It shows a sequence of produced bilingual tokens and corresponding labels in the introduced notation. The described dependency relations are extracted for a word *mSr* ('*Egyptian*') from our example in Figure 4.2.

4.3.3 Training

Training of dependency-based BiLMs consists of a sequence of extraction steps: After having produced word-alignments for a bitext (see Section 3.5.3), sentences are segmented according to Equation 4.2. We produce a dependency parse of a source sentence and a POS-tag labeling of a target sentence. For Chinese, we use the Stanford

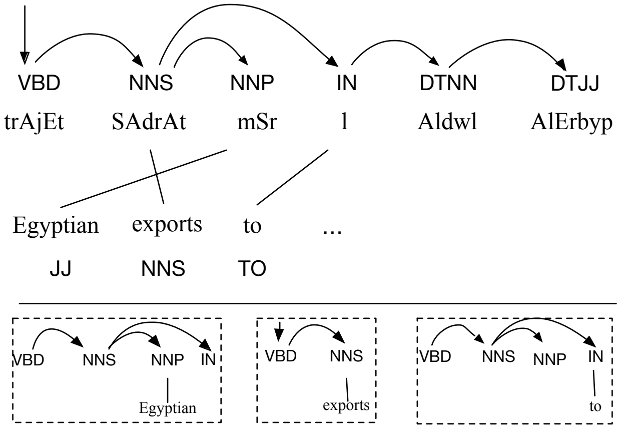


Figure 4.4: Sequence of bilingual tokens produced by a Pos→Pos→Pos+sibl•Lex after translating three words of the source sentence: VBD→NNS→ϵ+NNS+IN•Egyptian, ROOT→VBD→ ϵ+NNS+ϵ•exports, VBD→NNS→NNP+IN+ϵ•to (if there is no sibling on either of the sides, ϵ is returned).

dependency parser (Chang et al., 2009). For Arabic, a dependency parser is not available for public use, so we produce a constituency parse with the Stanford parser (Green and Manning, 2010) and extract dependencies based on the rules in (Collins, 1999). For English, POS-tagging, we use the Stanford POS-tagger (Toutanova et al., 2003). After having produced a labeled sequence of tokens, we learn a 5-gram model using SRILM (Stolcke et al., 2011). Kneyser-Ney smoothing is used for all model variations except for Pos•Pos where Witten-Bell smoothing is used due to zero count-of-counts.

4.3.4 Decoder integration

Dependency-based BiLMs are integrated into our phrase-based SMT decoder as follows: Before translating a sentence, we produce its dependency parse. Phrase-internal word-alignments, needed to segment the translation hypothesis into tokens, are stored in the phrase table, based on the most frequent internal alignment observed during training. Likewise, we store the most likely target-side POS-labeling for each phrase pair.

The decoding algorithm is augmented with one additional feature function and one additional, corresponding feature weight. At each step of the derivation, as a new phrase pair is added to the partial translation hypothesis, this function segments the new phrase into bilingual tokens (given the internal alignment information) and substitutes the words in the phrase pair with syntactic labels (given the source parse and the target POS labeling associated with the phrase). The new syntactified bilingual tokens are added to the stack of preceding $n-1$ tokens, and the feature function computes the weighted updated model probability. During decoding, the probabilities of the BiLMs are computed in a stream-based fashion, with bilingual tokens as string tokens, and not in a class-based fashion, with syntactic source-side representations emitting the corresponding target words (Bisazza and Monz, 2014).

4.4 Experiments

To evaluate the effectiveness of the proposed representations for BiLM tokens, we conduct a series of translation experiments. In each experimental run, we tune a model consisting of baseline features (see Section 3.5.3) and one of the dependency-based BiLM feature functions specified in Section 4.3.2. We compare the translation performance to a baseline PBSMT system and to a number of comparison systems that include BiLMs from (Niehues et al., 2011). The baseline features and the training, decoding and evaluation setups are described in Section 3.5.3. Throughout this chapter, we will use the name of a BiLM to refer to a translation system consisting of baseline features and this BiLM feature.

As comparison systems, we use BiLMs with two kinds of representations: lexical (Lex•Lex) and simple syntactic (Pos•Pos). The motivation behind the first one is to validate the assumption that local syntactic information is useful for SMT. If depBiLMs indeed show to outperform Lex•Lex, it will demonstrate that in such a constrained scenario whereby tokens are defined based on word positions syntax still helps to define meaningful models of translation. Specifically, this approach will help us answer **RQ1.c**: How do local syntactic representations compare to lexicalized representations? Comparing depBiLMs to Pos•Pos will help answer the question how much of the original syntactic representation should be incorporated into bilingual tokens. Pos•Pos represents the minimal amount of syntactic information. This will help us partially answer **RQ1.b**: How elaborate should the local representations be?

We must note the following imperfection in our experimental design: DepBiLMs can only be trained on the portion of a parallel corpus for which the source sentences have been parsed (an off-the-shelf parser may not find a well-structured parse for every input sentence). Lex•Lex training data does not require any annotation data, and Pos•Pos requires POS-tag annotation, which is a much simpler task than parsing. To account for this imbalance, one could only take the parsed portion of the training set also for training Lex•Lex and Pos•Pos. However, we did not do that. Tables 3.1 and 3.2 (Section 3.5.3) demonstrate the sizes of the used training corpora, as well as the sizes of the parsed subcorpora. We can conclude from it that depBiLM features were trained on a corpus more than 4.5 times smaller than the comparison BiLM features for Arabic-English, and almost 2.5 times for Chinese-English. Obviously, the disadvantage of this setting is that the comparison between the two features will not be totally fair. On the other hand, conceptually, this situation can be seen as "strengthened baselines", and if our proposed model manages to outperform the baseline and the comparison systems, it could provide stronger validation of the method's effectiveness. Another argument in favor is that lexicalized BiLM is likely to require more training data for good generalization, since its vocabulary is much larger. Thus, we could say we compensate for this asymmetry by differences in training data, and thus can compare the expressivity of the models (what kind of phenomena they capture) fairly. From a practical perspective, our experimental setup represents a realistic and very likely situation, where complex annotation such as parsing is often the bottleneck.

Just comparing lexical and POS-based BiLMs and depBiLMs with respect to a general-purpose metric will give a minimal answer to our main research question, which is whether depBiLMs improve reordering. In addition to that we would like to shed light as to whether depBiLMs help improve word-order related aspects of translation. Given our prior understanding of the models, Lex•Lex is likely to capture more local aspects of translation, while we expect the syntactically represented BiLMs to indeed improve reordering. Thus, we expect both models to provide complementary improvements. One way of verifying it is by combining them both in one system.

In the following subsections we discuss the general results for Arabic-English (Section 4.4.1) and Chinese-English (Section 4.4.2), where we use case-insensitive BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2009), and TER (Snover et al., 2006) as evaluation metrics. This is followed by a focused analysis of the models with respect to their reordering quality. The latter experiments include:

- evaluation with respect to reordering-sensitive metrics: n-gram precision with n of a high value, LRscore (Birch and Osborne, 2010) (Section 4.4.3);
- translation with an increased distortion limit (Section 4.4.4).

4.4.1 Arabic-English translation experiments

We present results of translation experiments in two formats: score evaluated on the concatenation of all of the available test sets (Table 4.1) and separate scores for each individual test set (Table 4.2 for BLEU, Table 4.4 for TER, Table 4.3 for METEOR).[4] Note that for each of the considered metrics the score on the concatenated set is not a linear function of the scores of the component sets. A bigger test set typically serves towards better estimation of the given metric and allows us to draw conclusions between systems with more confidence. We include separate scores for each test benchmark to allow for comparison with results in the literature.

We include results of randomization significance tests in the result tables. As explained in the preamble to Section 4.4, our points of comparison are the baseline, a system with Pos•Pos and a system with Lex•Lex. Statistical significance notation is explained in the caption of Table 4.1.

Overview. We first start with Table 4.1. From Table 4.1.a–b we can see that Lex•Lex generally yields significant improvements over the baseline, while Pos•Pos does not. This suggests that just the POS information of the words involved in reordering is too general to provide additional improvements. DepBiLM systems in Table 4.1.c-e significanly outperform the baseline (and Pos•Pos, which performs at the baseline level). This provides an indication that there needs to be a certain level of specificity in the representation of tokens in reordering. At the same time, we see that none of the depBiLM systems really outperforms Lex•Lex, except with respect to TER. We cannot take the result on TER to be a strong indication that depBiLM has better effect on translation, since the two other metrics with a diverse set of expertise[5] do not provide this indication.

[4] The test sets are: MT02, MT03, MT05, MT06, MT08, MT9. See Section 3.5.1 for more details.

[5] An exact matching BLEU and a semantic similarity matching METEOR, as opposed to just exact matching TER.

Table 4.1: BLEU, METEOR and TER scores for Arabic-English experiments evaluated on a concatination of all the test benchmarks (MT02, MT03, MT05, MT06, MT08, MT09). Note that since TER is an error rate, lower scores are better. **Statistical significance notation**: improvements are marked $^{\blacktriangle}$ at the $p < .01$ level and $^{\vartriangle}$ at the $p < .05$ level. Vertically reversed symbols ($^{\blacktriangledown}$ and $^{\triangledown}$) indicate statistically significant deterioration. $^{-}$ stands for no significant difference. For the systems in **(b)**, we only mark the difference with respect to the baseline (a). For each of the rest of the systems (**c–f**), the three symbols indicate, in this order: difference from baseline, Pos•Pos and Lex•Lex. If there is no improvement w.r.t. any of the comparison systems, we omit the annotation for this number.

	Configuration	**Arabic-English MT02-MT09 concatenated**		
		BLEU	METEOR	TER
a	PBSMT baseline	51.54	70.82	43.30
b	Pos•Pos	51.55	70.81	43.22 $^{\vartriangle}$
	Lex•Lex	52.07 $^{\blacktriangle}$	71.04	43.19 $^{\vartriangle}$
c	Pos→Pos•Pos	52.03 $^{\blacktriangle\blacktriangle-}$	71.11 $^{\blacktriangle\blacktriangle-}$	**42.92** $^{\blacktriangle\blacktriangle\blacktriangle}$
d	Pos→Pos−sibl•Pos	51.95 $^{\blacktriangle\blacktriangle-}$	70.94 $^{\vartriangle\vartriangle-}$	43.33 $^{--\triangledown}$
e	Pos→Pos→Pos•Pos	52.03 $^{\blacktriangle\blacktriangle-}$	71.10 $^{\blacktriangle\blacktriangle-}$	42.97 $^{\blacktriangle\blacktriangle\blacktriangle}$
f	Lex•Lex + Pos→Pos→Pos•Pos	**52.42** $^{\blacktriangle\blacktriangle\blacktriangle}$	**71.25** $^{\blacktriangle\blacktriangle\blacktriangle}$	43.05 $^{\blacktriangle\vartriangle\vartriangle}$

Lex•Lex vs depBiLMs. Our motivation for depBiLMs was based one the necessity of representations more general than lexical forms to capture word order phenomena. DepBiLMs do not outperform Lex•Lex on general-purpose metrics, and there are likely to be a few reasons for that. First, the baseline performance by itself is quite good and both kinds of BiLMs can further improve translation only by a small margin. Another reason is that the relatively dramatic difference in training data sizes does not allow depBiLM to achive the same degree of generalization. We ran an experiment whereby we added both the Lex•Lex and Pos→Pos→Pos•Pos[6] features into the system (Table 4.1.f). It demonstrates a significant improvement over all the comparison systems, which suggests that the two features yield complementary improvements. In Section 4.4.3 we take a closer look at what kind of aspects are improved.

Comparison between different depBiLMs. First of all, we can see that additional grandparent annotation does not appear to have an advantage over Pos→Pos•Pos (Table 4.1.c and .e). On the other hand, this additional grandparent specification does not lead to and deterioration (due to sparsity) and thus is a meaningful characterization of the

[6] We chose Pos→Pos→Pos•Pos since it gave the better improvements among depBiLMs and has a more complex representation than Pos→Pos•Pos, thus having better potential for learning more complex reordering patterns. See also the paragraph below.

Table 4.2: BLEU scores for Arabic-English experiments. Statistical significance notation is explain in the caption of Table 4.1.

Configuration	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**	**MT09**
PBSMT baseline	64.09	53.46	56.80	46.42	45.84	48.66
Pos•Pos	64.08	53.44	56.70	46.36	45.66	48.82
Lex•Lex	64.61 $^{\triangle}$	54.05 $^{\triangle}$	56.93	47.01 $^{\blacktriangle}$	46.19 $^{\blacktriangle}$	49.72 $^{\blacktriangle}$
Pos→Pos•Pos	64.80 $^{\blacktriangle\blacktriangle-}$	54.01 $^{\triangle\triangle-}$	57.10 $^{-\triangle-}$	46.82 $^{\triangle\blacktriangle-}$	46.03 $^{-\triangle-}$	49.48 $^{\blacktriangle\blacktriangle-}$
Pos→Pos−sibl•Pos	64.44	53.95 $^{-\blacktriangle-}$	57.00	46.55 $^{--\triangledown}$	45.85	49.12 $^{\triangle-\blacktriangledown}$
Pos→Pos→Pos•Pos	64.78 $^{\blacktriangle\blacktriangle-}$	54.19 $^{\triangle\triangle-}$	56.96	46.88 $^{\blacktriangle\blacktriangle-}$	45.93	49.37 $^{\blacktriangle\blacktriangle-}$
Lex•Lex + Pos→Pos→Pos•Pos	**65.23** $^{\blacktriangle\blacktriangle\blacktriangle}$	**54.50** $^{\blacktriangle\blacktriangle-}$	**57.31** $^{\triangle\blacktriangle\triangle}$	**47.36** $^{\blacktriangle\blacktriangle\triangle}$	**46.28** $^{\triangle\blacktriangle-}$	**50.08** $^{\blacktriangle\blacktriangle\triangle}$

Table 4.3: METEOR scores for Arabic-English experiments. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**	**MT09**
PBSMT baseline	75.72	74.27	75.95	67.37	67.24	69.62
Pos•Pos	75.61	74.27	75.99	67.41	67.10	69.71
Lex•Lex	75.82	74.32	76.01	67.64 $^{\blacktriangle}$	67.51 $^{\blacktriangle}$	69.99 $^{\blacktriangle}$
Pos→Pos•Pos	76.0 $^{\triangle\blacktriangle-}$	74.49	76.24 $^{\blacktriangle\triangle-}$	67.61 $^{\triangle--}$	67.49 $^{\triangle\blacktriangle-}$	70.07 $^{\blacktriangle\blacktriangle-}$
Pos→Pos−sibl•Pos	75.67	74.28	76.12	67.49	67.27	69.82
Pos→Pos→Pos•Pos	75.97 $^{\triangle\blacktriangle-}$	74.39	76.19 $^{\triangle--}$	67.74 $^{\blacktriangle\blacktriangle-}$	67.45 $^{-\blacktriangle-}$	70.01 $^{\blacktriangle\triangle-}$
Lex•Lex + Pos→Pos→Pos•Pos	**76.09** $^{\blacktriangle\blacktriangle\triangle}$	**74.54**	**76.35** $^{\blacktriangle\blacktriangle\blacktriangle}$	**67.78** $^{\blacktriangle\blacktriangle-}$	**67.65** $^{\blacktriangle\blacktriangle-}$	**70.22** $^{\blacktriangle\blacktriangle\triangle}$

Table 4.4: TER scores for Arabic-English experiments. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**	**MT09**
PBSMT baseline	37.45	41.74	38.56	46.20	47.23	43.93
Pos•Pos	37.45	41.64	38.40	46.10	47.26	43.75 $^{\triangle}$
Lex•Lex	37.36	41.63	38.61	46.14	47.21	43.54 $^{\blacktriangle}$
Pos→Pos•Pos	**36.82** $^{\blacktriangle\blacktriangle\blacktriangle}$	**41.25** $^{\triangle--}$	**38.35**	45.95 $^{\triangle--}$	**46.99**	43.35 $^{\blacktriangle\blacktriangle-}$
Pos→Pos−sibl•Pos	37.63	41.87	38.60	46.26	47.26	43.79
Pos→Pos→Pos•Pos	37.04 $^{\triangle\triangle-}$	**41.25** $^{\triangle--}$	38.38	**45.88** $^{\triangle-\triangle}$	47.12	43.37 $^{\blacktriangle\triangle-}$
Lex•Lex + Pos→Pos→Pos•Pos	37.04 $^{\triangle\triangle\triangle}$	41.30	38.42	46.11	47.22	**43.34** $^{\blacktriangle\blacktriangle-}$

reordering processes. At the same time, the sibling annotation (Table 4.1.d) does lead to a slight deterioration: the vocabulary of the generalization power provided by sibling information is not enough to salvage the sparsity issue resulting from the increased vocabulary.

Analysis of individual test sets' scores. We now look at the performance on individual test benchmarks (Tables 4.2–4.3). Lex•Lex yields significant improvements over the baseline for MT06-MT09, but not for MT02-MT05, and it is rather consistent across the metrics. This could be explained by our earlier suggestion that the baseline is already quite strong on *in-domain data* and is not easily improved. Table 4.5 contains information about genre distribution in the test sets. MT02-05 is entirely newswire data, a "native" genre for the Arabic-English trained system. However, the rest are a mixture of a few genres, where newswire is around 50 %. This observation suggests that Lex•Lex is able to capture patterns common across genres, giving up to 1 BLEU, 0.4 TER and 2 METEOR improvement over an individual test set. The picture for depBiLMs is less clear cut (with the exception of Pos→Pos−sibl•Pos, which demonstrates poor performance). We see improvements both for some (but not all) newswire-dominated sets and for some other genres as well, yielding up to 0.9 BLEU, 0.6 TER, 2 METEOR improvement over individual baseline scores. When Lex•Lex and Pos→Pos→Pos•Pos are combined together, the improvement is strongly statistically significant for almost all test sets/scores.

To summarize: The experiments demostrated that in general lexicalized BiLM Lex•Lex and (some) depBiLMs show comparable performance. There are some indications that they in fact give complementary contributions to translation quality, indiscernible by general purpose metrics. We also hypothesize that the baseline is already rather strong, thus also making it hard to compare performance of the two kinds of models. On top of it, Pos→Pos•Pos and Pos→Pos→Pos•Pos show very similar performance across test sets and metrics. In one of the subsequent sections we evaluate the models in a more unconstrainded search scenario (increasing the distortion limit), where the stronger expressive power of Pos→Pos→Pos•Pos can be tested.

Table 4.5: Distribution of genres across test benchmarks for Arabic-English. Genre labels are obtained from NIST documentation.

genre	MT02	MT03	MT05	MT06	MT08	MT09
newswire	100 %	100 %	100 %	42.5 %	60 %	44.5 %
broadcast	-	-	-	15 %	-	-
newsgroup	-	-	-	42.5 %	-	-
web	-	-	-	-	40 %	55.5 %

4.4.2 Chinese-English translation experiments

Like with the Arabic-English results, we present the systems' scores on a concatenated dataset (Table 4.6) and separately per test set (Tables 4.7–4.9). Also as before, we perform statistical significance testing of the Pos•Pos and Lex•Lex systems with respect to the baseline, and of the depBiLM systems with respect to the baseline, Pos•Pos and Lex•Lex.

Table 4.6: BLEU, METEOR and TER scores for Chinese-English experiments evaluated on a concatenation of all the test benchmarks (MT02, MT03, MT05, MT06, MT08). Statistical significance notation is explained in the caption of Table 4.1.

	Configuration	**Chinese-English MT02-MT08 concatenated**		
		BLEU	METEOR	TER
a	PBSMT baseline	31.68	59.14	58.76
b	Pos•Pos	31.89 ▲	59.22 △	58.73
	Lex•Lex	32.28 ▲	59.30 ▲	58.42 ▲
c	Pos→Pos•Pos	32.14 ▲▲−	59.43 ▲▲△	**57.96** ▲▲▲
d	Pos→Pos−sibl•Pos	32.0 ▲−▼	59.39 ▲▲−	58.07 ▲▲▲
e	Pos→Pos→Pos•Pos	32.72 ▲▲▲	**59.61** ▲▲▲	58.15 ▲▲▲
f	Lex•Lex + Pos→Pos→Pos•Pos	**32.77** ▲▲▲	59.58 ▲▲▲	58.36 ▲▲−

Overview. First of all, we see that all of the BiLM variants improve the baseline. In general, the picture is quite different from the one for Arabic-English, as elaborated below. We connect this difference to the difference in baseline quality (the Arabic-English training set is much larger) and the relative differences between the full training set and the depBiLM training set. For Arabic-English, the training set is 4.5 times larger, while for Chinese-English it is 2.5. Consequently, the relative effect of depBiLM is larger for the latter. Unfortunately, the given experimental setup does not allow us to connect the observed differences to typological differences between Arabic and Chinese.

Lex•Lex vs depBiLMs. Again in contrast to the results for Arabic-English, depBiLMs produce statistically significant improvements over the strongest comparison system Lex•Lex. As discussed before (Section 4.4), the relative differences in training sizes for depBiLMs and Lex•Lex is smaller for this language pair than for Arabic-English. Thus, when the respective training sizes are close, depBiLMs are shown to have a stronger impact on translation quality. The question still remains whether the two models capture different translation phenomena, or depBiLM just subsumes the expressive power of Lex•Lex. The combined system in Table 4.6.f suggests that the 'expertises' of the two models are different, since it gives no improvements over Pos→Pos→Pos•Pos.

Comparison between different depBiLMs. The depBiLM feature with sibling annotation (Table 4.6.d) demonstrates the worst performance, like for Arabic-English. Grandparent annotation improves translation, as demonstrated in Table 4.6.c and .e. We hypothesize that reordering dependencies are characterized by longer spans for

Table 4.7: BLEU scores for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**
PBSMT baseline	33.31	33.55	33.64	32.59	25.93
Pos•Pos	33.61 $^{\triangle}$	33.79 $^{\triangle}$	33.82	32.77 $^{\triangle}$	26.00
Lex•Lex	34.01 $^{\blacktriangle}$	34.12 $^{\blacktriangle}$	34.20 $^{\blacktriangle}$	33.50 $^{\blacktriangle}$	26.36 $^{\blacktriangle}$
Pos→Pos•Pos	33.85 $^{\triangle--}$	34.53 $^{\blacktriangle\blacktriangle\triangle}$	34.05 $^{\triangle--}$	33.07 $^{\triangle-\triangledown}$	26.19
Pos→Pos−sibl•Pos	33.88 $^{\triangle--}$	34.65 $^{\blacktriangle\blacktriangle-}$	33.66 $^{--\triangledown}$	32.49 $^{--\triangledown}$	26.58 $^{\blacktriangle\blacktriangle-}$
Pos→Pos→Pos•Pos	34.28 $^{\blacktriangle\blacktriangle-}$	**35.07** $^{\blacktriangle\blacktriangle\blacktriangle}$	**34.58** $^{\blacktriangle\blacktriangle\blacktriangle}$	33.59 $^{\blacktriangle\blacktriangle-}$	26.77 $^{\blacktriangle\blacktriangle\triangle}$
Lex•Lex + Pos→Pos→Pos•Pos	**34.37** $^{\blacktriangle\blacktriangle-}$	34.98 $^{\blacktriangle\blacktriangle\blacktriangle}$	34.38 $^{\blacktriangle\blacktriangle-}$	**33.74** $^{\blacktriangle\blacktriangle-}$	**27.03** $^{\blacktriangle\blacktriangle\blacktriangle}$

Table 4.8: METEOR scores for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**
PBSMT baseline	60.71	59.91	61.39	59.58	54.79
Pos•Pos	60.82	60.04	61.59 $^{\blacktriangle}$	59.67	54.70
Lex•Lex	**60.90**	59.86	61.66 $^{\triangle}$	59.87 $^{\blacktriangle}$	54.88
Pos→Pos•Pos	60.85	60.08	61.79 $^{\triangle--}$	59.93 $^{\blacktriangle\triangle-}$	55.13 $^{\triangle\blacktriangle\triangle}$
Pos→Pos−sibl•Pos	60.9	60.30 $^{\triangle-\blacktriangle}$	61.68	59.76	55.06 $^{\triangle\triangle-}$
Pos→Pos→Pos•Pos	60.85	**60.56** $^{\blacktriangle\blacktriangle\blacktriangle}$	**61.86** $^{\blacktriangle--}$	**60.21**	55.18 $^{\blacktriangle\blacktriangle\triangle}$
Lex•Lex + Pos→Pos→Pos•Pos	60.89	60.30 $^{\blacktriangle-\blacktriangle}$	61.76 $^{\triangle--}$	60.18 $^{\blacktriangle\blacktriangle\blacktriangle}$	**55.32** $^{\blacktriangle\blacktriangle\blacktriangle}$

Table 4.9: TER scores for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**
PBSMT baseline	59.50	58.72	59.09	57.03	60.04
Pos•Pos	59.46	58.60	58.95	57.03	60.13
Lex•Lex	58.97 $^{\blacktriangle}$	58.28 $^{\triangle}$	58.73 $^{\triangle}$	56.81	59.77 $^{\triangle}$
Pos→Pos•Pos	58.92 $^{\triangle\triangle-}$	**57.88** $^{\blacktriangle\blacktriangle\triangle}$	**58.02** $^{\blacktriangle\blacktriangle\blacktriangle}$	**56.33** $^{\blacktriangle\blacktriangle\blacktriangle}$	59.24 $^{\blacktriangle\blacktriangle\blacktriangle}$
Pos→Pos−sibl•Pos	**58.71** $^{\blacktriangle\blacktriangle-}$	57.96 $^{\blacktriangle\blacktriangle-}$	58.39 $^{\blacktriangle\triangle-}$	56.54 $^{\blacktriangle\blacktriangle-}$	**56.21** $^{\blacktriangle\blacktriangle\blacktriangle}$
Pos→Pos→Pos•Pos	59.35	58.04 $^{\blacktriangle\blacktriangle-}$	58.55 $^{\blacktriangle\triangle\triangle}$	56.28 $^{\blacktriangle\blacktriangle\blacktriangle}$	59.27 $^{\blacktriangle\blacktriangle\blacktriangle}$
Lex•Lex + Pos→Pos→Pos•Pos	59.80 $^{--\blacktriangledown}$	58.11 $^{\blacktriangle\triangle-}$	58.75	56.7 $^{\triangle\triangle-}$	**59.21** $^{\blacktriangle\blacktriangle\blacktriangle}$

Table 4.10: Distribution of genres across test benchmarks for Chinese-English. Genre labels are obtained from NIST documentation.

genre	MT02	MT03	MT05	MT06	MT08
newswire	38 %	100 %	100 %	37 %	51 %
broadcast	-	-	-	34 %	-
newsgroup	-	-	-	29 %	-
web	-	-	-	-	49 %
speech	62 %	-	-	-	

Chinese-English than for Arabic-English, and this may partially explain why grandparent specification is necessary for the former language pair.

Analysis of individual test sets' scores. Also for each individual test set most of the considered BiLMs outperform the baseline. However, we observe a somewhat non-uniform performance of Pos→Pos→Pos•Pos vs Lex•Lex. For MT03 and MT05, which fully consist of newswire (Table 4.10), depBiLMs outperforms Lex•Lex. Additionally, the combination of the two features deteriorates the performance of Pos→Pos→Pos•Pos alone. On the other "mixed genre" test sets, the advantage of depBiLM is much less pronounced, and the combination of the two features tends to give additional improvements. We think that a likely explanation is that on a new domain the baseline performs worse and the additional expertise of the lexicalized BiLM is necessary.

To summarize, we observed that for Chinese-English all of the BiLM systems that we ran outperform the system. Pos→Pos→Pos•Pos demonstrates the best results among all of the depBiLMs. It also tends to outperform BiLMs. We also saw that on partially out-of-domain test sets both Lex• Lex and Pos→Pos→Pos•Pos actually both contribute to an increase in quality (like for Arabic-English).

In the subsequent sections of this chapter we will look more closely at what kind of phenomena are captured by the different models, which may provide more insights into the results for general-purpose metrics. In Section 4.4.3 we evaluate translation output from Sections 4.4.1 and 4.4.2 with respect to reordering-sensitive metrics. In Section 4.4.4 we run the systems from Sections 4.4.1 and 4.4.2 in a decoding setting with an increased distortion limit.

4.4.3 Reordering-sensitive evaluation metrics

In order to answer **RQ1**, whether depBiLMs improve reordering, we evaluated the subset of better performing systems with respect to 4-gram precision and LRscore. We present scores on the concatenated test sets (Tables 4.11 and 4.14), and on individual test sets (Tables 4.12, 4.13, 4.15, 4.16).

4-gram precision (prec$_4$) is a component of BLEU. It is the share of n-grams in the translation output that matches an n-gram in the references (see Equation 2.22 in Section 2.3.1). This metric is suitable to estimate short-distance reordering, with the constraint of exact lexical matching. If a system scores high with prec$_4$, it indicates

Table 4.11: 4-gram precision and LRscore for Arabic-English experiments evaluated on a concatenation of all the test benchmarks (MT02, MT03, MT05, MT06, MT08, MT09).

	Configuration	**Arabic-English MT02-MT09 concatenated**	
		prec$_4$	LRscore
a	PBSMT baseline	32.30	0.6644
	Lex•Lex	32.86	0.6636
c	Pos→Pos•Pos	32.79	**0.6646**
e	Pos→Pos→Pos•Pos	32.79	0.6644
f	Lex•Lex + Pos→Pos→Pos•Pos	**33.25**	0.6642

Table 4.12: 4-gram precision for Arabic-English PBSMT baseline and BiLM pipelines.

Configuration	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	44.92	33.47	37.95	27.64	26.56	30.15
Pos•Pos	44.77	33.41	37.74	27.63	26.36	30.38
Lex•Lex	45.61	34.21	37.99	28.23	26.90	31.10
Pos→Pos•Pos	**45.77**	34.14	38.16	28.03	26.69	31.16
Pos→Pos→Pos•Pos	45.72	34.39	37.94	28.16	26.64	31.10
Lex•Lex + Pos→Pos→Pos•Pos	46.37	**34.79**	**38.43**	**28.47**	**26.99**	**31.65**

Table 4.13: LR scores scores for Arabic-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explain in the caption of Table 4.1.

Configuration	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	0.6409	0.6636	0.6873	0.6621	0.6636	**0.6688**
Lex•Lex	**0.642**	0.6657	0.6854	0.6610	0.6619	0.6658
Pos→Pos•Pos	0.6418	0.6642	0.6874	0.6616	**0.6647**	0.6682
Pos→Pos→Pos•Pos	0.6411	0.666	0.6872	**0.6622**	0.664	0.6661
Lex•Lex + Pos→Pos→Pos•Pos	0.6419	**0.6685**	**0.6882**	0.6592	0.6622	0.6655

Table 4.14: 4-gram precision and LRscore for Chinese-English experiments evaluated on a concatenation of all the test benchmarks (MT02, MT03, MT05, MT06, MT08).

	Configuration	**Chinese-English**	
		MT02-MT08 concatenated	
		prec$_4$	LRscore
a	PBSMT baseline	14.03	0.4853
b	Lex•Lex	14.52	0.4838
c	Pos→Pos•Pos	14.43	**0.4866**
d	Pos→Pos→Pos•Pos	14.79	0.4861
e	Lex•Lex + Pos→Pos→Pos•Pos	**14.88**	0.4845

Table 4.15: 4-gram precision for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explain in the caption of Table 4.1.

Configuration	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	13.96	15.00	15.34	14.79	11.06
Pos•Pos	14.19	15.18	15.41	14.79	11.06
Lex•Lex	14.63	15.49	15.65	15.37	11.47
Pos→Pos•Pos	14.52	15.95	15.34	15.10	11.38
Pos→Pos→Pos•Pos	14.80	16.18	15.88	15.50	11.69
Lex•Lex + Pos→Pos→Pos•Pos	**14.91**	**16.28**	**15.69**	**15.56**	**12.04**

Table 4.16: LR scores scores for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explain in the caption of Table 4.1.

Configuration	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	0.5076	0.4839	0.49	0.4825	0.4627
Lex•Lex	0.5076	0.4840	0.4860	0.4805	0.4609
Pos→Pos•Pos	0.5110	0.4861	0.4893	0.4844	0.462
Pos→Pos→Pos•Pos	0.5100	0.4869	0.4901	0.4830	0.4605
Lex•Lex + Pos→Pos→Pos•Pos	0.5078	0.4865	0.4881	0.4817	0.4585

that it is able get the right order of words locally, and that it is good at picking correct lexical formulations. If a system does not fare well with prec$_4$, then it could be that the model actually gets the correct order, but fails to choose the right word translation. Or it could be that the system indeed fails at local reordering (but could still do well on a more global level.

For Arabic-English (Table 4.11), all of the BiLM systems improve prec$_4$ of the baseline. Lex•Lex slightly outperforms the rest of the systems, and the combined system in Table 4.11.e gives the best results. This is the same pattern as observed for the general-purpose metrics. We can conclude that both depBiLMs and lexicalized BiLM improves local short-distance reordering in complementary ways. Results on individual tests (Table 4.12) are mostly consistent with the above findings.

For Chinese-English (Table 4.14), the picture is similar to Arabic-English, except that Pos→Pos→Pos•Pos shows the best performance in this case. The combined system in Table 4.14.e shows the absolute best score, indicating that both kinds of BiLM improve short distance reordering for Chinese-English. This is also consistent with results on individual tests (Table 4.15).

LRscore was specifically designed to measure reordering quality of output translation. Section 2.3.1 provides details on the hyperparameters and computation of this metric, while we here we only summarize the version of the metric that we use here:

$$LRscore(\mathcal{A}_{\mathit{ref}}, \mathcal{A}_{\mathit{trans}}) = \frac{\mathit{Hamming}(\mathcal{A}_{\mathit{ref}}, \mathcal{A}_{\mathit{trans}}) + (1 - \mathit{KendallTau}(\mathcal{A}_{\mathit{ref}}, \mathcal{A}_{\mathit{trans}}))}{2}, \tag{4.3}$$

where $\mathcal{A}_{\mathit{ref}} : F \to \mathit{Ref}$ is word alignment between source and reference interpreted as function from source sentence F into *Ref*, and $\mathcal{A}_{\mathit{trans}} : F \to \mathit{Trans}$ is word alignment between source and the given translation. LRscore interprets alignments as permutations of the source sentence and computes the average of the Hamming distance and the inverse Kendall Tau distance for every sentence (the higher the metric value, the better). The final metric is the average of the sentence set. As opposed to prec$_4$, it does not rely on exact lexical matching, and by design it characterizes any type of reordering (of any distance). Its disadvantage is that it highly depends on the quality of the word alignments between source and translation/reference.

For Arabic-English (Table 4.11), all of the systems, including the baseline, show very similar performance in terms of LRscore. This result suggests that Arabic-English does not have long-distance reordering patterns. It also could be that LRscore is too noisy to capture more subtle differences and local reordering patterns, which we saw for prec$_4$.

For Chinese-English (Table 4.14), the LRscore does differentiate between the different systems. depBiLM systems obtain the best scores, and lexicalized BiLM the worst. The combination of grandparent annotated depBiLM and lexicalized BiLM performs worse than the depBiLM alone.

To summarize, comparing performance of BiLMs across language pairs and reordering sensitive metrics, we can make a tentative conclusion that Lex•Lex is good at capturing short-distance (local) reordering. DepBiLMs are shown to be good at both short and long distance reordering, with the more richly annotated variant (Pos→Pos→Pos•Pos) showing better results. For Arabic-English, however, we did not obtain experimental confirmation of the existence of long distance reordering: it either is not typical for this language pair, or is already captured by the baseline, or too hard to capture by any of the considered models.

4.4.4 Decoding with an increased distortion limit

So far, we only looked at translation output produced with a constraint that a reordering of maximum five words is allowed during decoding. It is a very limiting assumption, since due to the property of linguistic recursion, the reordering distances can in principle be made very large. In this subsection we relax this constraint to 10 and 15 word jumps. This relaxation, however, entails challenges for the translation systems: the search space increases considerably, and the models should still be able to rank good translation hypotheses high.

Note that we use the models tuned with a distortion limit of 5. Arguably, this could be a limitation to the experiment, but the presented results still can be regarded as the lower bound of the performance of the models in an extended search space scenario. The results on a concatenated set are presented in Tables 4.17 and 4.18, and on individual test sets in Tables 4.19-4.24.

Arabic-English systems (Table 4.17) demonstrate overall deterioration of results with an increased distortion limit (number in brackets are the differences from the original result with a distortion limit of 5). The most dramatic deterioration is observed for Pos→Pos•Pos. The best results are again obtained for the combination of two BiLMs, but still the scores are lower than for the scenario with a distortion limit of 5. We can conclude that it is likely that Arabic-English does not typically have long distance reordering and/or that at least the systems that we worked with cannot generalize beyond short distance reordering.

Chinese-English systems (Table 4.18) present quite the opposite situation. All of the systems increase their scores when the distortion limit goes up. This is a strong indication that the (near) optimal translation hypotheses are contained in the extended search space and that long distance reordering is typical for this language pair. The optimum seems to be closer to the distortion limit of 10, rather than 15, as the table demonstrates. Pos→Pos→Pos•Pos produces the best results, and the combination of two BiLMs deteriorates them to some extent. These findings are consistent with our observations in the previous section, where we saw that Pos→Pos→Pos•Pos is good at long distance reordering, but not Lex•Lex.

4.5 Conclusions

In this chapter, we have introduced a simple, yet effective way to include syntactic information into phrase-based SMT. Our method consists of enriching the representation of units of a bilingual language model (BiLM). We argued that the very limited contextual information used in the original bilingual models (Niehues et al., 2011) can capture reorderings only to a limited degree and proposed a method to incorporate information from a source dependency tree in bilingual units. In a series of translation experiments

Table 4.17: BLEU, METEOR, TER scores of Arabic-English systems with increased distortion limit (DL) during decoding on the concatenation of all the test sets (MT02, MT03, MT05, MT06, MT08). The trained systems are the same as in previous chapters (tuned with standard distortion limit of 5). The numbers in brackets indicate relative difference to the score obtained by decoding with a distortion limit of 5. Statistical significance notation is the same as in previous experiments and is explained in the caption of Table 4.1.

	Configuration	DL	**Arabic-English** **MT02-MT09 concatenated**		
			BLEU	METEOR	TER
a	PBSMT baseline	10	51.34 (-0.2)	70.80 (+0.02)	43.56 (+0.26)
		15	51.13 (-0.41)	70.70 (-0.12)	43.92 (+0.62)
b	Lex●Lex	10	51.50 (-0.57)	70.88 (-0.07)	43.81 (+0.59)
		15	51.11 (-0.96)	70.75 (-0.15)	44.54 (+1.32)
c	Pos→Pos●Pos	10	49.81 (-2.22)	70.30 (-0.79)	44.92 (+ 2)
		15	49.71 (-2.32)	70.24 (-0.85)	45.19 (+2.27)
d	Pos→Pos→Pos●Pos	10	51.35 (-0.68)	70.87 (0.23)	**43.00** (+0.03)
		15	51.19 (-0.84)	70.82 (-0.28)	43.31 (+0.34)
e	Lex●Lex + Pos→Pos→Pos●Pos	10	**51.72** (-0.7)	**71.08** (-0.17)	43.82 (+0.77)
		15	51.15 (-1.27)	70.86 (-0.39)	44.78 (+1.73)

Table 4.18: BLEU, METEOR, TER scores of Chinese-English systems with increased distortion limit (DL) during decoding on the concatenation of all the test sets (MT02, MT03, MT05, MT06, MT08). The trained systems are the same as in previous chapters (tuned with standard distortion limit of 5). The numbers in brackets indicate relative difference to the score obtained by decoding with a distortion limit of 5. Statistical significance notation is the same as in previous experiments and is explained in the caption of Table 4.1.

	Configuration	DL	**MT02-MT08 concatenated**		
			BLEU	METEOR	TER
a	PBSMT baseline	10	32.14 (+0.46)	59.3 (+0.84)	58.95 (-0.19)
		15	31.85 (+0.17)	59.24 (+0.1)	59.73 (+0.97)
	Lex•Lex	10	32.72 (+0.44)	59.53 (+0.1)	58.51 (+0.55)
		15	32.46 (+0.18)	59.42 (+0.12)	59.15 (+1.19)
c	Pos→Pos•Pos	10	33.00 (+0.86)	59.82 (+0.39)	**57.73** (-0.23)
		15	32.87 (+0.73)	59.74 (+0.31)	58.06 (+0.1)
e	Pos→Pos→Pos•Pos	10	**33.47** (+0.75)	**59.96** (+0.35)	58.21 (+0.06)
		15	33.13 (+0.41)	59.83 (+0.22)	58.89 (0.74)
f	Lex•Lex + Pos→Pos→Pos•Pos	10	33.25 (+0.58)	59.74 (+0.16)	58.62 (+0.26)
		15	32.94 (+0.17)	59.60 (+0.02)	59.56 (+0.2)

Table 4.19: BLEU scores for Arabic-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explain in the caption of Table 4.1.

Configuration	DL	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**	**MT09**
PBSMT baseline	10	64.18	53.38	56.78	45.86	**45.65**	48.40
	15	63.96	53.26	56.65	45.60	45.48	48.12
Lex•Lex	10	64.29	53.64	56.49	46.18	45.57	48.90
	15	63.82	53.18	56.23	45.77	45.02	48.62
Pos→Pos•Pos	10	61.67	51.17	54.37	44.83	43.35	**49.38**
	15	61.68	51.14	54.35	44.70	43.24	49.18
Pos→Pos→Pos•Pos	10	**65.09**	**54.64**	**57.10**	45.66	44.20	48.62
	15	65.01	54.39	56.95	45.51	43.98	48.45
Lex•Lex + Pos→Pos→Pos•Pos	10	64.51	53.95	56.85	**46.42**	45.62	49.24
	15	64.16	53.55	56.40	45.79	44.74	48.70

Table 4.20: METEOR scores for Arabic-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explain in the caption of Table 4.1.

Configuration	DL	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**	**MT09**
PBSMT baseline	10	75.8	74.25	76.02	67.21	67.2	69.61
	15	75.67	74.22	75.93	67.08	67.12	69.5
Lex•Lex	10	75.82	74.25	75.95	67.34	67.31	69.79
	15	75.73	74.06	75.86	67.22	67.07	69.69
Pos→Pos•Pos	10	74.77	73.34	75.15	66.85	66.48	**70.1**
	15	74.81	73.3	75.19	66.78	66.36	70.0
Pos→Pos→Pos•Pos	10	**76.12**	74.48	76.17	67.21	66.8	69.79
	15	76.08	74.44	76.04	67.15	66.79	69.75
Lex•Lex + Pos→Pos→Pos•Pos	10	76.02	**74.49**	**76.19**	**67.47**	**67.49**	70.02
	15	75.92	74.37	76.07	67.18	67.13	69.79

Table 4.21: TER scores for Arabic-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	DL	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**	**MT09**
PBSMT baseline	10	37.48	41.71	38.52	46.76	47.51	44.30
	15	37.82	42.06	38.83	47.17	47.87	44.68
Lex•Lex	10	37.80	42.00	38.97	46.95	47.98	44.20
	15	38.41	42.47	39.37	47.72	48.99	45.05
Pos→Pos•Pos	10	39.71	43.83	40.60	48.12	49.58	**43.43**
	15	39.84	43.95	40.70	48.44	49.94	43.83
Pos→Pos→Pos•Pos	10	**36.93**	**40.92**	**38.15**	**46.09**	**47.23**	43.56
	15	37.13	41.27	38.42	46.41	47.56	43.92
Lex•Lex + Pos→Pos→Pos•Pos	10	37.54	41.72	38.81	47.19	48.15	44.21
	15	38.33	42.27	39.53	48.24	49.43	45.19

Table 4.22: BLEU scores for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	DL	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**
PBSMT baseline	10	33.80	33.81	34.46	33.16	25.98
	15	33.66	32.93	34.14	32.97	25.89
Lex•Lex	10	34.61	34.44	34.84	33.81	26.65
	15	34.31	33.95	34.62	33.59	26.55
Pos→Pos•Pos	10	34.81	35.21	34.92	34.06	26.86
	15	34.70	34.62	34.90	34.06	26.77
Pos→Pos→Pos•Pos	10	**35.37**	**35.33**	35.21	**34.54**	**27.44**
	15	35.02	34.49	**35.25**	34.21	26.85
Lex•Lex + Pos→Pos→Pos•Pos	10	35.05	35.31	34.87	34.46	26.92
	15	34.84	34.52	34.71	34.10	26.78

Table 4.23: METEOR scores for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explained in the caption of Table 4.1.

Configuration	DL	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**
PBSMT baseline	10	61.1	60.06	61.87	59.73	54.58
	15	60.89	59.79	61.77	59.78	54.65
Lex•Lex	10	**61.24**	60.08	62.13	59.99	54.89
	15	61.12	59.95	61.94	59.94	54.83
Pos→Pos•Pos	10	**61.24**	60.61	62.13	60.36	55.38
	15	61.1	60.32	62.12	60.31	55.42
Pos→Pos→Pos•Pos	10	61.4	**60.84**	**62.23**	**60.54**	**55.45**
	15	61.3	60.45	62.3	60.38	55.29
Lex•Lex + Pos→Pos→Pos•Pos	10	61.12	60.51	62.12	60.36	55.16
	15	61.07	60.05	62.06	60.21	55.13

Table 4.24: TER scores for Chinese-English PBSMT baseline and BiLM pipelines. Statistical significance notation is explain in the caption of Table 4.1.

Configuration	DL	**MT02**	**MT03**	**MT05**	**MT06**	**MT08**
PBSMT baseline	10	59.26	59.44	58.90	57.01	60.67
	15	60.44	60.44	59.46	57.66	61.38
Lex•Lex	10	59.25	58.47	58.50	56.68	60.18
	15	59.95	59.51	59.10	57.34	60.52
Pos→Pos•Pos	10	**58.49**	**57.80**	**57.71**	**55.94**	**59.26**
	15	58.95	58.41	57.97	56.06	59.62
Pos→Pos→Pos•Pos	10	59.11	58.43	58.33	56.40	59.46
	15	59.78	59.35	58.63	57.08	60.30
Lex•Lex + Pos→Pos→Pos•Pos	10	59.75	58.65	58.76	56.67	60.03
	15	60.76	59.89	59.69	57.62	60.70

we performed a thorough comparison between various syntactically-enriched BiLMs and competing models. The results demonstrated that adding syntactic information from a source dependency tree to the representations of bilingual tokens in an n-gram model can yield statistically significant improvements over the competing systems. Even though trained on a smaller sub-corpus than comparison BiLMs, depBiLMs could achieve the same or better level of generalization. A number of additional evaluations provided an indication for better modeling of reordering phenomena. When evaluated on reordering-sensitive metrics, we found that depBiLMs indeed improve reordering, while lexicalized BiLMs tend to preform better at short-distance reordering. In experiments with an increased distortion limit, we found that all Arabic-English systems deteriorate translation quality, but systems with both a depbiLM and lexicalized BiLM features combined do so to a lesser extent. As for Chinese-English, the distortion limit of 10 (during testing) produced best results overall, with parent- and grandparent-annotated BiLMs giving the best result.

We can now revisit and answer the research question **RQ1** raised at the beginning of the chapter. We will answer each of the subquestions separately:

RQ1 Can we improve reordering by modeling sequences of syntactic structures representing basic operational units of translation?

RQ1.a. Can the representations only include the local syntactic information of a node in a syntactic parse? What is the minimum context that the local representation should incorporate?

DepBiLMs are BiLMs with local syntactic representations, defined in terms of the immediate vicinity of a node in a dependency tree. Our extensive experiments showed that depBiLMs improve translation quality overall, and reordering in particular, as demonstrated by reordering-sensitive metrics and translation experiments with an increased distortion limit. We compared the translation performance of depBiLMs to BiLMs with simple POS-based representations. The latter showed some of the worst results, only barely improving the baseline. From this we can conclude that there should be a certain degree of specificity in the syntactic representation to improve translation. Among depBiLMs, both parent-annotated (Pos→Pos•Pos) and parent- and grandparent-annotated (Pos→Pos→Pos•Pos) depBiLMs demonstrated significant improvements over the baseline in general and with respect to reordering. Sibling-annotated depBiLM showed mixed performance, not always even outperforming the baseline. This suggests that sibling information is too specific and does not always provide good generalization of the token's syntactic and reordering behavior.

RQ1.b. How do local syntactic representations compare to representations including explicit lexical information of the basic translational units?

In general, depBiLMs produced translations at least as good as the one produced by lexicalized BiLMs. We have found some indications that they in fact capture complementary phenomena (at least for Arabic-English), whereby Lex•Lex is better at more specific and short-term reordering, while depBiLMs capture more general patterns of translation correspondence.

We also note that even though depBiLMs were trained on a substantially smaller training corpus, they were able to achieve the same or better level of generalization (which is expected given the smaller and more abstract vocabulary).

RQ1.c. What kind of reordering phenomena are captured by such models?

As we mentioned before, depBiLMs seem to be good at reordering in general. We could find evidence of this model to perform especially well at some specific subclass of reordering patterns. But given the abstract definition of the depBiLM representations, it is expected that it would be a "universal" reordering model, while more fine-grained lexicalized BiLMs capture better reordering phenomena for specific classes of words.

While in this chapter we have shown the usefulness of local syntactic characterization of minimal bilingual translation units, in the following chapter we explore a method that imposes syntactic structure on the translation sequence in a more intricate way. Instead of just using the source syntax for labeling, it derives a new target syntactic structure that is used in a syntactic language model.

Finally, an interesting question is, can local syntactic representations also be useful in neural MT. A number of papers have recently shown the usefulness global syntactic encoding in NMT (Shi et al., 2016; Eriguchi et al., 2016). Local source syntactic information has also been recently exploited in the context of neural MT. Bastings et al. (2017) propose to substitute the standard RNN-based encoder of the sequence-to-sequence model with a graph-convolutional encoder. The convolution filter in their model is applied to to sets of words determined by their vicinity in a dependency tree. This is very much in the spirit of our depBiLM representations incorporating the immediate siblings and parents of a word in a tree.

5

Bilingual Structured Language Models for Statistical Machine Translation

5.1 Introduction

In the previous chapter we used syntax to construct representations for translation units involved in a strictly sequential, syntax-agnostic decoding process. The function of these representations was to define a vocabulary capable of capturing important contextual properties of the translation units. These representations were strictly constrained by the process of translation hypothesis derivation.

In this chapter we take a step away from purely sequential models and define a sequential model constrained by a hierarchical structure. In Chapter 3 we reviewed a number of approaches that integrate hierarchical models into the phrase-based translation framework. One subclass of such methods define constraints on the phrase-based translation process. For example, Cherry (2008) defines soft constraints based on the notion of syntactic cohesion. Ge (2010) captures reordering patterns by defining soft constraints based on the currently translated word's POS tag and the words structurally related to it. Defining translation constraints based on the analysis of the source sentence is advantageous since the latter can be made available prior to translation. On the other hand, target syntax is more challenging to use in PBSMT, since a target-side syntactic model does not have access to the whole target sentence at decoding time (Post and Gildea, 2008; Schwartz et al., 2011). Post and Gildea (2008) is one of the few target-side syntactic approaches applicable to PBSMT, but it has been shown not to improve translation. Their approach uses a target side parser as a language model. One of the reasons why it fails is that a parser assumes its input to be grammatical and chooses the most likely parse for it. A syntactic model is likely to be useful if it says how good the translation hypothesis actually is.

This chapter is the continuation of our research on integrating source syntax into PBSMT. In contrast to the models from the literature mentioned above that define constraints on the translation hypothesis search, we propose a weaker model in the sense that it does not simply impose a constraint on the translation search, but produces structural representations of the target sentence conditioned on the translation derivation. When constructing the model, we rely on a fundamental assumption about the nature of the correspondence between source and target sentence structures in a parallel sentence pair, which is the expressed in **RQ2**:

RQ2 Is there a systematic mapping between source and target syntactic representations in a parallel sentence and can it be used to improve translation?

We compare two approaches to finding and evaluating a mapping between source and target structures. The first one is expressed by **RQ2.a**:

RQ2.a. Is there a universal characterization of a mapping between source and target structure? Can this characterization be used to constrain the decoding process to produce better translations?

We answer **RQ2.a** by implementing an existing constraint-based model (syntactic cohesion). Our contribution in this chapter is another approach to discovering the systematic mapping in question, expressed in **RQ2.b**:

RQ2.b. Can the mapping be defined in terms of projection constraints between *elementary* parts of source and target structures? Can we fit a statistical model over the resulting corresponding source and target structures to characterize the overall mapping?

To address these questions, we start with a set of possible constraints on how source substructures should be mapped to target substructures. Different constraint combinations implicitly define a set of mappings. We learn a characterization of each mapping by fitting a statistical model over pairs of source and target sentence structures derived based on the direct correspondence assumption. The potential advantage of learning a model over the resulting representations instead of directly imposing constraints is that the statistical model can learn non-trivial patterns of interactions between the constraints. Finally, we evaluate what kind of constraints are useful:

RQ2.c. What are the important mapping constraints that result in structured language models improving translation output?

As a statistical model of parsed target sentences, we adapt an existing monolingual model of Chelba and Jelinek (2000), namely *structured language model* (SLM), to the bilingual setting. We use this model because it is simple and has been shown to work as a characterization of parsed sentences before. We should note that SLMs have been incorporated into a translation system before. Yamada and Knight (2001) use SLMs in a string-to-tree SMT system where a derivation of a target-side parse tree is part of the decoding algorithm, and target syntactic representations are obtained 'for free'. Yu et al. (2014) use an on-the-fly shift-reduce parser to build an incremental target parse. Our approach is different in that our general translation framework is purely sequential and the target parses are obtained without the need for additional probabilistic inference. This property of our method makes it attractive from a practical point of view, allowing for easier implementation and only a minor increase in runtime complexity.

Our contributions in this chapter can be summarized as follows:

1. We compare our approach of characterizing a correspondence between source and target structures with a statistical language model to previous research on the question of how systematic the correspondence is (Section 5.2).

2. We propose a novel method to adapt monolingual structured language models by Chelba and Jelinek (2000) (Section 3.4) to a PBSMT system (Section 5.3), which does not require an external on-the-fly parser, but only uses the given source-side syntactic analysis to infer structural relations between target words. We refer to this model as bilingual structured language models (**BiSLMs**).

3. Building on the existing literature, we propose a set of deterministic rules that incrementally build up a parse of a target translation hypothesis based on the source parse (Section 5.3). We discuss a few variations of the set of rules, which we take to be a hyperparameter of our BiSLM.

4. We evaluate our method in a series of rescoring experiments and achieve statistically significant improvements in BLEU for Chinese-English, but not Arabic-English (Section 5.4.2). We use the rescoring experiments to understand how useful the structured language model is for translation when the word alignment is fixed. Additionally, based on the rescoring experiments we identify best-performing BiSLM variants and run an experiment with the BiSLM fully integrated into the decoder (Section 5.4.3). In the latter experiments we achieve statistically significant improvements for both Arabic-English and Chinese-English, for all the considered metrics.

5. We experimentally compare our approach to the baseline without any syntactic knowledge incorporated and a system with syntactically defined soft constraint features (Section 5.4.1).

5.2 Direct correspondence assumption

In this section we discuss the idea of crosslingual syntactic correspondence, which underlies this chapter, and its different formulations proposed in the literature.

We first fix a syntactic formalism that is used throughout the chapter. Just like in the previous chapter, we take a dependency tree *Tree*$_W$ to be a syntactic representation of sentence W and reason about other syntactic assumptions and models in its terms. A dependency tree *Tree*$_W$ induces a dependency relation D between words of W (where $D(w, w')$ means w is a parent of w'). We choose a dependency structure over a constituency structure because the former is more general, as every constituency parse can be formalized as a projective dependency parse with labeled relations, but not vice versa (Osborne, 2008). Moreover, also for the sake of simplicity, we assume *unlabeled* dependency trees. Finally, we make a projectivity[1] assumption, which is supported by empirical data in many languages (Kuhlmann and Nivre, 2006; Havelka, 2007), and makes our model computationally less expensive. For more details about the dependency formalism and its properties see Section 3.1.

Most NLP models that use syntactic structures to model the interaction between two or more languages rely on some form of the *direct correspondence assumption* (DCA) (Hwa et al., 2002). It makes a statement about a correspondence between the syntactic

[1] Dependency tree *Tree*$_W$ and the induced relation D are projective if $\forall w_i, w_j, w_k \in W : (D(w_i, w_j) \wedge w_k \textit{ between } w_i \textit{ and } w_j) \rightarrow (D(w_i, w_k) \vee D(w_j, w_k))$.

structures of the source and target sides across parallel sentences. We first provide the full original statement of DCA, and discuss its linguistic (empirical) motivations below. Just like in this thesis, Hwa et al. (2002) formulate statements in terms of dependency structures, since the choice of the most general possible formalism gives more chance of this statement being valid empirically. The direct correspondence assumption is defined as follows:

> Given sentences *E* and *F* that are (literal) translations of each other with respective syntactic structures Tree_E and Tree_F: If nodes x_E and y_E of Tree_E are aligned[2] with nodes x_F and y_F of Tree_F, respectively, and if syntactic relationship $D(x_E, y_E)$ holds in Tree_E, then $D(x_F, y_F)$ holds in Tree_F.

This statement calls for two comments: First, we stress that it is about literal translations. Therefore the degree of applicability of this assumption depends on the quality of the given parallel corpus. Second, the employed concept of *alignment* is not tied to a specific word alignment algorithm. Rather, it is the ideal notion of word alignment which is something like semantic or translational equivalence between words in two languages. In this general statement above alignment is assumed to be an injective relation, but of course in the real linguistic world, there are no two languages for which this property can hold fully. The whole agenda of empirical verification of DCA and its application in NLP tasks is about bridging the discrepancy between the idealized notion of alignment and the practically observable alignment obtained based on specific data and software.

DCA is grounded linguistically, as literal translation equivalents are likely to express the same set of thematic relations (Hwa et al., 2002). Since dependency relations are supposed to be the formalization of thematic relations conveyed by the sentences, it follows that translationally equivalent sentences also have the same set of dependency relations. At the same time, every natural language provides its own way of formalizing semantics (in terms of syntax), so in practice the equivalence between the sets of dependency pairs will be approximate at best. There is ample empirical evidence supporting the violation of DCA even in literal translation (Hwa et al., 2002). In practice, also the degree of literariness of translation will vary, exacerbating the issue.

Hwa et al. (2002) conduct a series of experiments aimed at answering the question: To what degree is DCA true? They also pose a second question, but do not directly evaluate it: How useful is DCA? This chapter is an attempt to research the second question in the context of machine translation, as our method consists in deriving representations for a structured language model, with our starting assumption point being DCA. The DCA has been used before in downstream NLP tasks. There are models of cross-lingual transfer that define syntactic structure of one language by conditioning it on the structure of semantically equivalent sentences in another language (Hwa et al., 2005; Naseem et al., 2012). DCA has also been used in SMT. In particular, syntax-based SMT is built implicitly around this assumption (Wu, 1997b; Yamada and Knight, 2001). In (Quirk and Menezes, 2006) DCA is explicitly implemented by defining a translation model in terms of treelet pairs where target-side treelets are produced by projecting

[2] In the sense of translation correspondence.

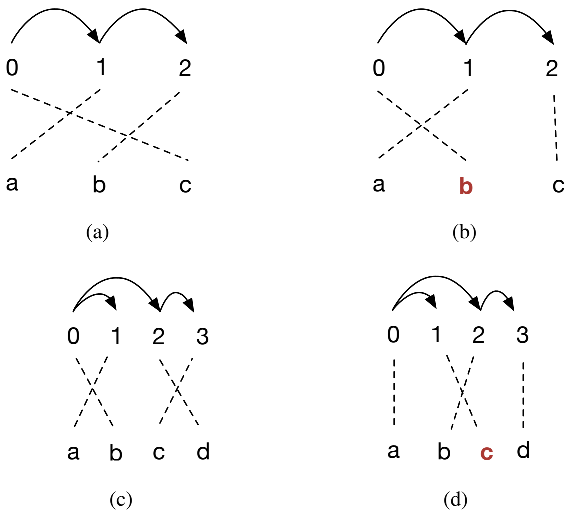


Figure 5.1: Examples of cohesive and uncohesive translations. (a-b): cohesive (a) and uncohesive (b) translations of the same dependency parse. (b) is uncohesive because words a and c translate the source subtree $\{(1, 2)\}$, but the target word b does not translate this subtree. (c-d): cohesive (c) and uncohesive (d) translations. (d) is uncohesive because a and c translate the source subtree $\{(0, 1)\}$, but b does not translate it.

source dependencies via word alignments. Moreover, they use the projected syntactic parse on the target side to define syntax-based language models, very much in the spirit of our approach here.

5.2.1 Weaker forms of DCA and their use in machine translation

DCA in its original form is a very strong assumption and therefore may not be borne out in practice. However, the intuition behind it is linguistically motivated and weaker forms of this assumption could be made directly useful in machine translation.

Syntactic cohesion (Fox, 2002; Cherry, 2008) is a constraint that does not allow for non-projective reordering: Given a source parse $Tree_F$ and the induced relation D_F, a translation E is *cohesive* if all translated target words e_i, e_j do not have any word e_k between them such that there is a source subtree sub in $Tree_F$ such that some parts of it are translated by w_i and w_j but not by w_k (see examples in Figure 5.1). Syntactic cohesion is a weaker assumption than DCA in the formal sense: it allows a greater class of mutually corresponding source and target syntactic structures, of which the ones defined by DCA is a subclass (given that we assume projectivity of syntactic structures). Cherry (2008) and Bach et al. (2009) implement the syntactic cohesion assumption as a set of soft constraints applied during phrase-based decoding. Their implementation of cohesion is not strict in that they only require phrase applications, and not necessarily individual target words, to conform to the cohesion principle. For example, if we imagine a situation where a subtree as in Figure 5.1(b) is translated as a whole with one phrase application (and not word by word), then it does not violate the cohesion principle, although it is internally uncohesive.

We use the set of features from (Bach et al., 2009) as a comparision system in our experiments, so we summarize their model here. The feature **coh1** is the original model from (Cherry, 2008). It is a constraint that is evaluated to truth if at the current translation step i it is the case the previous translation step $i-1$ a source subtree was translated and it is not covered by the current phrase application. **coh2** is the constraint, but it applies to all preceding phrase applications. **coh3** and **coh4** represent the same constraints as **coh1** and **coh2**, respectively, but the corresponding features are as the number of untranslated words in the unfinished subtrees. **coh5** does the same as **coh4**, but only for words that have the part of speech *Noun* or *Verb*.

Research question **RQ2.b** suggests another weaker hypothesis in the spirit of DCA: there exists *some* correspondence between source and target structures and one can fit a model to characterize it. This is the idea behind this chapter. We learn a language model that scores sequences obtained as a result of some mapping between source and target trees, that we induce based on a set of simple rules. This model can then be used at prediction time to score sequences obtained via decoding, thus implicitly evaluating the reordering quality of obtained translation hypotheses.

5.3 Bilingual structured language models

In this section, we combine a weak form of the direct correspondence assumption (Section 5.2.1), positing that there is some systematic correspondence between source and target structures, and structured language models (SLMs) (Section 3.4), and define bilingual structured language models (BiSLMs) for PBSMT.

As we said before, this is not the first time SLMs are adapted to SMT. However, it is the first time—to the best of our knowledge—that it is applied to a syntactically agnostic framework (PBSMT). Previous approaches (Yamada and Knight, 2001; Yu et al., 2014; Quirk and Menezes, 2006) rely on resources that a standard PBSMT system does not have access to by default. Phrase-based decoders do not provide us with a parse of the target sentence, and inferring the parse of a target string with an external parser is computationally expensive and potentially unreliable (see Section 7.1). Our main insight is that in a bilingual setting one does not need an additional probabilistic target parsing model.

SLM (see Section 3.4) models generation of parsed sentence as a sequence of n steps of picking a new word to add to the sentence and then deciding how to integrate it with the partial tree:[3]

[3] This is a simplified version of the original model, where there is an intermediate step of predicting the POS: $p_{SLM}(W, Tree^W) = \prod_{i=1}^{|W|} p(w_i|pos_i, Expos(W_{i-1}, Tree^W_{i-1})) \cdot p(pos_i|Expos(W_{i-1}, Tree^W_{i-1})) \cdot p(Tree^W_i|w_i, Expos(W_{i-1}, Tree^W_{i-1}))$.

$$\begin{aligned} p_{SLM}(W, \mathit{Tree}^W) = \prod_{i=1}^{|W|} & p(w_i|\mathit{Expos}(W_{i-1}, \mathit{Tree}^W_{i-1})) \\ & \cdot p(\mathit{Tree}^W_i|w_i, \mathit{Expos}(W_{i-1}, \mathit{Tree}^W_{i-1})), \end{aligned} \tag{5.1}$$

where W_{i-i} is a partial sentence $w_1...w_i$, Tree^W_{i-1} is the subtree structure covering W_{i-i}, and *Expos* returns a conditioning context based on the partial parse so far. The second factor in the equation is the implicit notation for prediction of a parse operation (shift, right-/left-reduce), by which the tree is extended.

We assume that there is systematic mapping between source and target structures, but unlike the previous class of approaches outlined above, we do not assume that there is a systematic mapping between source and target structures that can be characterized with one simple property that holds universally. Rather, we propose to decompose this mapping into elementary projection steps of basic source structures onto basic target structures. Since we are working with unlabeled dependency trees, the basic structure is an unlabeled dependency arc. Thus, we assume there exists a function $\mathit{ProjP}(\mathit{Tree}^F, \mathcal{A}, F, E_i)$ that outputs Tree^E_i (a subtree structure of the partial target sentence E_i) based in source sentence Tree^F, source and target pair F and E and the alignment $\mathcal{A}$ between them. In this chapter we assume *ProjP* to be deterministic, therefore the second factor in Equation 5.1 is not needed. Thus we obtain the formula for a bilingual structured language model (BiSLM):[4]

$$p_{BiSLM}(E|F, \mathcal{A}, \mathit{Tree}^F) = \prod_{i=1}^{|E|} p(e_i|\ \mathit{Expos}(E_{i-1}, \mathit{ProjP}(\mathit{Tree}^F, \mathcal{A}, F, E_i))), \tag{5.2}$$

In words, at each time step i we predict the next word e_i conditioned on the exposed heads of the partial parse of E_{i-1} projected from the source side. We limit *Expos* to returning the four preceding exposed heads.

We start with a preliminary example in Figure 5.3 to illustrate how the model works. Since word alignment is monotonic in Figure 5.2(a), it is straightforward to project the source dependencies onto the target side. We aim to imitate a monolingual parser in the way we build up our projected parse: Reduce operations should be invoked whenever both of the subtrees involved in the operation are not expected to have any more modifiers (Section 5.3.2). For example, when the target word *likes* is produced its exposed heads are *said* and *he* (Figure 5.2(b)), since *Putin* is a modifier of *said*. Likewise, the exposed heads for *women* are *said likes all Russian* (Figure 5.2(c)).

Now the question is, how should *ProjP* be defined? In an ideal scenario, we would like to learn *ProjP* itself from data. The conclusion of Section 5.2 was that we would not like to come up with and impose an ad-hoc correspondence relation, but define it as a variable and learn its characterization. In this chapter we consider a few projection constraints (with a few variations) inspired by previous research and treat their different

[4] In case *ProjP* was statistically parametrized, the formula would look like: $p_{BiSLM}(E|F, \mathcal{A}, \mathit{Tree}^F) = \prod_{i=1}^{|E|} p(\mathit{ProjP}(\mathit{Tree}^F, \mathcal{A}, F, E_i))p(e_i|\ \mathit{Expos}(E_{i-1}, \mathit{ProjP}(\mathit{Tree}^F, \mathcal{A}, F, E_i)))$.

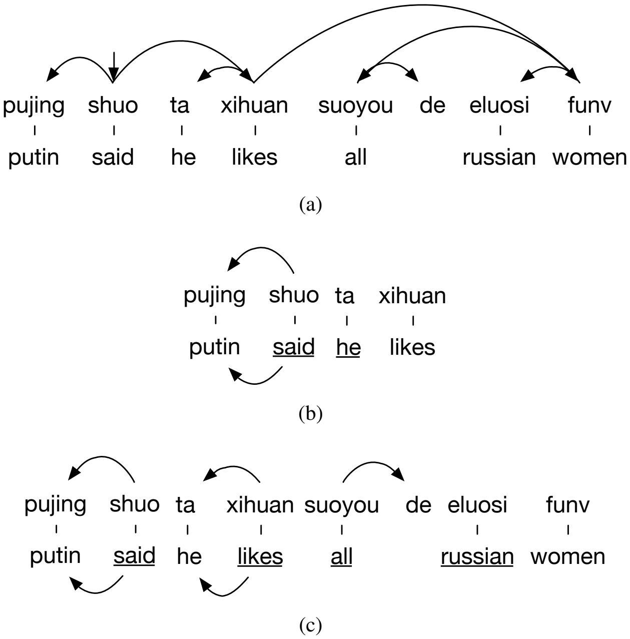

pujing shuo ta xihuan suoyou de eluosi funv
putin said he likes all russian women
(a)
pujing shuo ta xihuan
putin said he likes
(b)
pujing shuo ta xihuan suoyou de eluosi funv
putin said he likes all russian women
(c)

Figure 5.2: Chinese-English sentence pair (a) and sets of exposed heads (underlined) at different generation (b and c) steps of a bilingual SLM.

combinations as a hyperparameter of the model. The projection strategies are evaluated in a series of rescoring experiments (see Section 5.4.2 below). Since evaluating all possible projection strategies is computationally prohibitive, we propose a basic set of rules and three parameters of variation. One direction of future work would be to evaluate these parameters during training of the BiSLM (for example, via expectation-maximization) and not at the stage of model selection.

As opposed to projection approaches like those of Quirk and Menezes (2006), we would like our model to project a source parse incrementally, allowing it to be used in a PBSMT decoder. We think of *ProjP* as a function that computes the output in two stages: First, it infers from the source parse the dependency relations between target words (Section 5.3.1). This can be seen as setting constraints on how target words should be connected via dependency relations. Second, it decides how to actually parse the target sequence, i.e., in which order to assign these dependencies (Section 5.3.2). This can be seen as a constraint imposed by the sequential process of target generation.

Additionally, in Section 5.3.3 we propose to use additional labeling strategies of target words, and in Section 5.3.4 we describe estimation and implementation details.

5.3.1 Dependency graph projection

Adoption of some form of DCA (Section 5.2) allows us to build up a target dependency tree from a source tree by projecting the latter through word alignments. The definition of DCA can be rephrased as requiring a one-to-one correspondence *mapping* between words of a sentence pair, allowing one to unambiguously map dependencies: Given a source parse, if t_1 is the head of t_2, then $\mathit{map}(t_1)$ is the head of $\mathit{map}(t_2)$. The correspondence relation that we have in PBSMT is the word alignment *align*: in the most general case, it is a many-to-many correspondence, and the straightforward projection described above can lead to incorrect dependency structures. To overcome these problems, we describe a simple ordered set of projection rules, based on the ones specified by Quirk and Menezes (2006) (and we point out if otherwise).

The general idea behind this set of rules is to extract a one-to-one function $\boldsymbol{align}_{1-1}$ from source words to target words from *align* and use it to project source dependencies as described in the paragraph above (and **R1** below). The definition of align_{1-1} essentially describes how to remove alignment links so that the resulting mapping is one-to-one. Intuitively, align_{1-1} defines those source words which are the most important in determining the projected target structure. We decided to prioritize left-hand alignment links in multi-aligned sets of words, which is an arbitrary decision and alternatives (arguably, equally arbitrary) are conceivable.[5] align_{1-1} in the combination with rule **R1** represent the idealized projection algorithm. The additional rules (**R2-R4** below) are designed to project arcs which connect source words not in align_{1-1} or to incorporate target words not in align_{1-1}.

A function can be seen as a set of pairs, and so we build up the set align_{1-1} incrementally, by considering each of $\langle e_1, ..., e_n \rangle = E$ one by one, from left to right. We define a "partial" set $\mathit{align}_{1-1}^{\mathbf{part}_i}$ as follows:

$$\mathit{align}_{1-1}^{\mathbf{part}_0} = \varnothing \tag{5.3}$$

$$\mathit{align}_{1-1}^{\mathbf{part}_i} = \begin{cases} \mathit{align}_{1-1}^{\mathbf{part}_{i-1}} \cup \{(f, e_i)\}, & \text{if } (f, e_i) \in \mathit{align} \\ & \wedge \neg \exists f' : (f', e_i) \in \mathit{align} \wedge f' \prec f \\ & \wedge \neg \exists e' : (f, e') \in \mathit{align}_{1-1}^{\mathbf{part}_{i-1}} \\ \mathit{align}_{1-1}^{\mathbf{part}_{i-1}}, & \text{otherwise,} \end{cases} \tag{5.4}$$

where $f' \prec f$ means that f' linearly precedes f in a sentence. Finally, align_{1-1} is defined as $\mathit{align}_{1-1}^{\mathbf{part}_n}$ obtained at the last processing step n (which is the length of the English sentence E). This algorithm may leave some English words unaligned with respect to align_{1-1}.

For example, in Figure 5.4(b) the link between f_0 and e_1 is not in align_{1-1}, and in Figure 5.4(c) the link between f_1 and e_0 is removed (and therefore the arc from f_2 to f_1 does not get projected).

The following list of rules specify what dependency structure gets assigned to a target sequence, given the source dependency structure and the alignment. We call them rules, as they have the form *if X, then Y*: however, they function as constraints in the parsing procedure described in Section 5.3.2. They are ordered, so that the next rule can be applied in case the rules are not satisfied. For every e_i, e_j:

(R1) if $\exists f_k,\ f_l : D_F(f_k, f_l) \ \wedge\ (f_k, e_i) \in \mathit{align}_{1-1} \wedge\ (f_l, e_j) \in \mathit{align}_{1-1}$, then

[5] For example, Devlin et al. (2014) keep the middle alignment link.

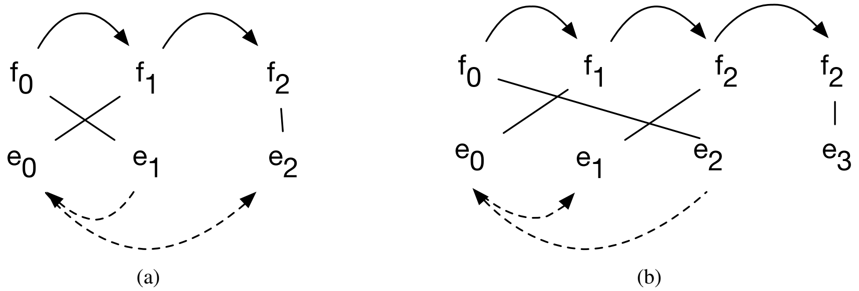


Figure 5.3: (a): The dashed lines are the dependency arcs that would project through word alignment, resulting in a non-projective projective (impossible under strong source-completeness). (b): The dashed lines are the parse produced under weak source-completeness. Under strong completeness none of the words will get connected.

$D_E(e_i, e_j)$

(R2) if $\exists f \in F : (f, e_i) \in \mathit{align}_{1-1} \ \wedge \ (f, e_j) \in \mathit{align}$, then $D_E(e_i, e_j)$

(R3) if $\exists f_k,\ f_l : D_F(f_l, f_k) \ \wedge \ (f_k, e_i) \in \mathit{align}_{1-1} \wedge \ (f_l, e_j) \in \mathit{align} \ \wedge e_i \prec e_j$, then $D_E(e_j, e_i)$

(R4) if $\exists f_k,\ f_l : D_F(f_k, f_l) \ \wedge \ (f_k, e_i) \in \mathit{align} \wedge \ (f_l, e_j) \in \mathit{align}_{1-1} \ \wedge e_i \prec e_j$, then $D_E(e_i, e_j)$

(R5) if $\neg\exists f : (f, e_i) \in align_{1-1}$, then we have two alternative rules:

(a) $\neg\exists e : D_E(e_i, e) \ \vee \ D_E(e, e_i)$

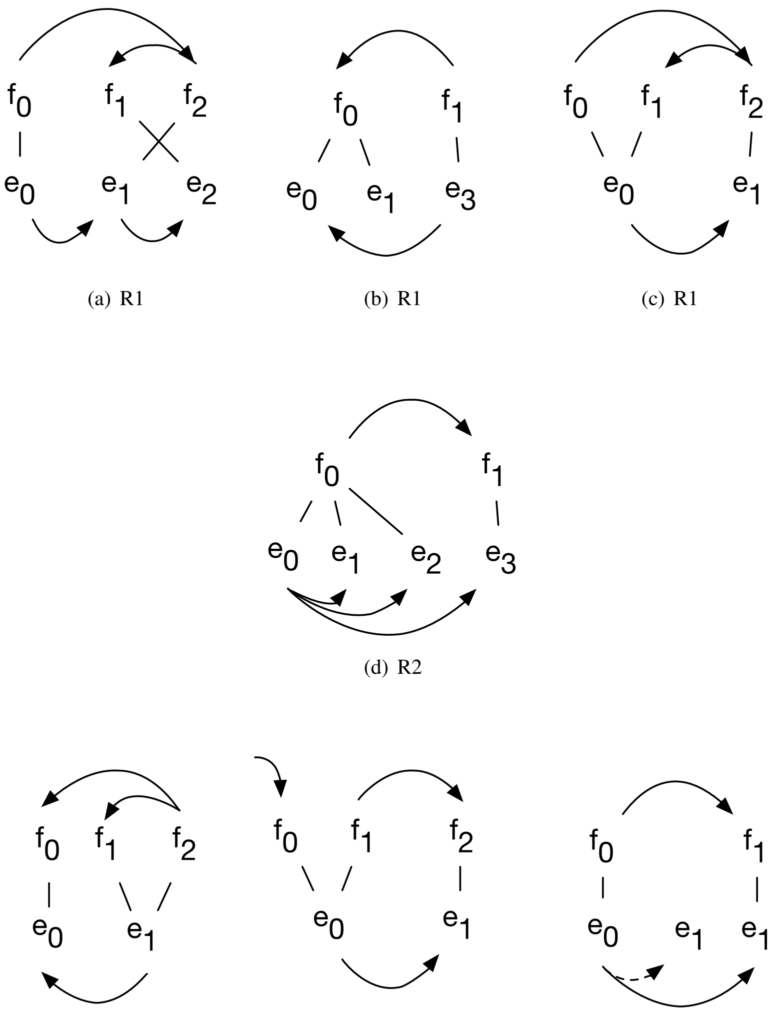
(b) $D_E(e, e_i)$, where e is the root of the preceding subtree on the stack (since these rules are integrated into the parsing procedure specified in Section 5.3.2)

Figure 5.4 contains examples of how *align*$_{1-1}$ is constructed and of the applications of the rules above. We note that some of the rules (namely, R1, R2 and R5) are applied in different contexts, so their mutual ordering does not actually matter. However, the mutual ordering of R1, R3, R4 matters.

5.3.2 BiSLM parsing procedure

Given an inference procedure for dependency relations between target words (Section 5.3.1), we should further specify in which order the corresponding dependency arcs are assigned to the target sentence as it is being generated.

We define an incremental parsing procedure in terms of three operations: *shift*, *left-reduce*, and *right-reduce* (see Section 3.1 for details on incremental dependency parsing). The operations are applied as soon as the sufficient conditions hold. We specify the conditions using the following structural properties:



(a) R1

(b) R1

(c) R1

(d) R2

(e) R3

(f) R4

(g) R5

Figure 5.4: Examples for dependency projection rules. **(a)**: all aligned word pairs are in *align*$_{1-1}$, R1 applies. **(b)**: $(f_0, e_1$ link is not in *align*$_{1-1}$, R1 applies to e_0 and e_3, as they are in *align*$_{1-1}$. **(c)**: (f_1, e_0) link is not in *align*$_{1-1}$, rule R1 applies to (f_0, e_0) and (f_2, e_1), which are in *align*$_{1-1}$. **(d)**: e_1 and e_2 are not in the range of *align*$_{1-1}$, therefore they get adjoined to e_0 by R2. **(e)**: e_0 and e_1 cannot be connected by R1, but since (f_2, e_1) is in *align*, and $D_F(f_2, f_0)$, they get connected by R3. **(f)**: e_0 and e_1 cannot be connected by R1, there is an arc from f_1, which is in the domain of *align*, to f_2, so they get connected by R4. **(g)** demonstrates two versions of R5: the dashed arrow gets 'realized' only if we adjoin unaligned words to the preceding head.

> A target subtree is *source-complete* if all the descendants of $align_{1-1}^{-1}(root(sub))$ (source correspondent of the root of the current subtree) (Section 5.3.1) have been translated and reduced. A target subtree is *complete* if it is source-complete and all the target words that are its children through non-projected arcs (through R2 or R4 in Section 5.3.1) have been translated and reduced.

The bilingual parsing operations and the sufficient conditions for them are defined as follows:

- **Shift:** after the word is produced it is shifted onto the stack as an elementary subtree.
- **Left-reduce**: if a disconnected subtree sub_i and a disconnected subtree sub_{i-1} immediately preceding it are both complete and $D_T(root(sub_i), root(sub_{i-1}))$, adjoin sub_{i-1} to sub_i so that $root(sub_{i-1})$ is a modifier of $root(sub_i)$.
- **Right-reduce**: analogous to *left-reduce*, but $D_T(root(sub_{i-1}), root(sub_i))$.

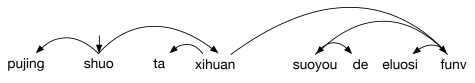
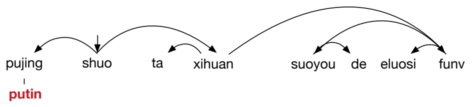
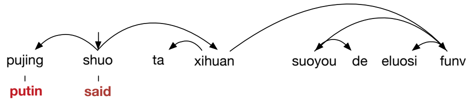
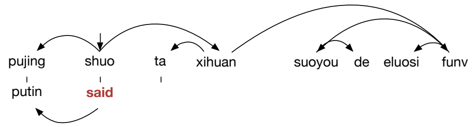
Training data may contain instances of non-cohesive translations. In that case the resulting target dependencies are non-projective. Our definition of left- and right-reduce only produces projective parses. For a non-cohesive translation, certain subtrees will never be source-complete and will never be reduced; see Figure 5.3(a). Note that this is not a disadvantage of our model. Cherry (2008) simply assumes that non-cohesive reordering should be penalized, and our model is able to account for this pattern. We therefore consider an alternative to incorporating non-cohesive alignments by relaxing the definition of completeness for subtrees:

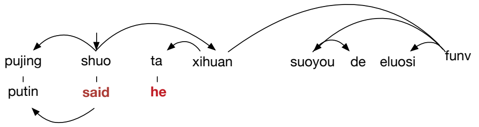
> A projected subtree sub is *weakly source-complete* if all descendants of all source word(s) which are aligned to the root of sub have been translated and, *only* if the definition of reduce applies, reduced; see Figure 5.3(b).

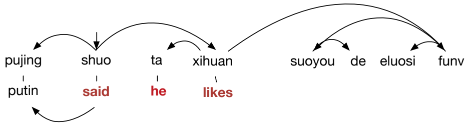
5.3.3 Syntactic labeling of tokens

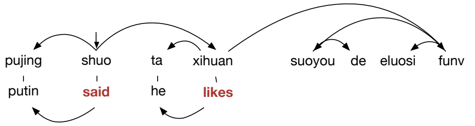
One of the problems with SLMs in general is that at time steps i and j the sets of exposed heads for t_i and t_j can differ in size, which may imply different predictive power. To this end, we add an additional detail to our model: Each time a reduction occurs, we label the root of the subtree to which another subtree has been adjoined, thus making the conditioning history more specific. We use the following labelings:

- **Reduction labeling:** if a subtree is adjoint to *sub* from the left, then label *root(sub)* with **LR**. If it is adjoint from the right, then label it with **RR**.
- **Reduction POS-labeling:** same as in simple reduction labeling, but add the POS tag of the root of the reduced subtree to the label.

operation	parsed parallel sentence	resulting stack	strucLM probability of the stack
-		<s>	p(<s>)
shift		<s> putin	p(<s>) p(putin)
shift		<s> putin said	p(<s>) p(putin) p(said|putin)
reduce-left		<s> said ↓ putin	(same as above)

shift	 pujing shuo ta xihuan suoyou de eluosi funv
	putin said he
		<s> said he
		↓
		putin
			p(<s>) p(putin) p(said|putin)
			p(he|said)

shift	 pujing shuo ta xihuan suoyou de eluosi funv
	putin said he likes
		<s> said he likes
		↓
		putin
			p(<s>) p(putin) p(said|<s> putin)
			p(he|<s> said) p(likes|<s> said he)

reduce-left	 pujing shuo ta xihuan suoyou de eluosi funv putin said he likes	<s> said likes ↓ ↓ putin he	(same as above)

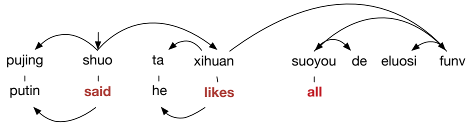
shift	 pujing shuo ta xihuan suoyou de eluosi funv
	putin said he likes all
		<s> said likes all
		↓ ↓
		putin he
			p(<s>) p(putin) p(said | <s> putin)
			p(he | <s> said) p(likes | <s> said he)
			p(all | <s> said likes)

Figure 5.5: Example of inference with a BiSLM (with no reduction labeling). This is a simplified example since it does not contain unaligned words or uncohesive translation. Its purpose is to demonstrate the basic mechanism of the approach.

5.3.4 Implementation and training

To use a BiSLM during decoding or rescoring, one needs access to phrase-internal alignments and target POS tags. We store phrase-internal alignments and target-side POS annotations of each phrase in the phrase table, based on the most frequent internal alignment during training and the most likely target-side POS labeling $\hat{t}$ given the phrase pair: $\hat{t} = \arg\max_{\bar{t}} p(\bar{t}|\bar{e}, \bar{f})$ (Monz, 2011). We train a BiSLM on the parallel training data (Section 3.5.3) and use the Stanford dependency parser Chang et al. (2009) for Chinese and the Stanford constituency[6]and parser (Green and Manning, 2010) for Arabic. POS-tagging of the training data is produced with the Stanford POS-tagger (Toutanova et al., 2003).

We estimate BiSLM probabilities with an n-gram model with modified Kneser-Ney smoothing using SRILM (Stolcke et al., 2011). The order is set to 5 (i.e., the number of exposed heads returned by *Expos* from Equation 5.2 is 4).

To summarize, in this section we proposed a way to adapt a structured language model to a translation setting, where the target parse is deterministically inferrable from the source parse (Equation 5.2). We proposed a series of rules that identify elementary dependency relations on the target side based on the source parse and word alignments (Section 5.3.1). We propose a sequential left-to-right projection method (Section 5.3.2). Figure 5.5[7] provides an example of how target structure is built up incrementally and how probability of the partial sequence is computed.

Overall the projection method has three variation parameters:

1) how unaligned target words are incorporated into target parse (rule R4): adjoining it to an immediately preceding head, or leaving it disconnected;

2) whether we allow for weak completeness when doing parsing reduction operation;

3) whether we employ reduction labeling – this parameter has three values: no labeling, reduction labeling, reduction-POS labeling.

5.4 Experiments

In this section we conduct a series of experiments to answer **RQ2**: whether there is a systematic relation between source and target sentence structures in parallel sentence. We evaluate it by comparing two approaches that tentatively assume **RQ2**: the first one simply predefines a relation between syntactic structures in a parallel sentence, the second one induces a target structure based on a given source structure via a set of elementary rules and learns a syntactic language model that characterizes the resulting target structures. We compare the two approaches by incorporating them into a PBSMT translation system and evaluating their effect on translation quality.

We propose the following set of experiments: First of all, we start with a baseline translation experiment not containing any of the syntax-based features discussed in this chapter (see Section3.5.3 for the description of our baseline PBSMT system). Further,

[6] We extract dependency parses from its output based on Collins (1999)

[7] We thank Zhaochun Ren for helping to design this example.

we test the assumption that the notion of syntactic cohesion of translation can be successfully used to constrain the search space and select better translation hypotheses. We do this by including all of the five proposed features from Bach et al. (2009), which include the original soft constraint from Bach et al. (2009); see Section 3 for a detailed description of the features. We optimize the weights of these features during tuning alongside the rest of the baseline features. These experiments are presented in Section 5.4.1.

Next, we present two sets of experiments designed to test our proposed method. In Section 5.4.2 we start with rescoring experiments where we use the BiSLM to rerank the n-best translation list output produced by the baseline system. The rescoring experimental setup keeps the alignments obtained as a result of translation search fixed, thus allowing us to test the usefulness of structured language models in relative isolation. It allows us to answer the question whether the choice of the syntactic model is good enough as an application of (some weak form of) DCA. Moreover, we use the rescoring experiments as a means of choosing the optimal projection strategy (see Section 5.3 for a description of projection strategies). Naturally, a rescoring experiment has its limitations, namely a very restricted hypothesis set. Additionally in our case, it also comes with low confidence of whether alignments inside parallel sentences are good since we have no guarantee that the n-best hypotheses produced by the baseline system are actually good translations.

As we will see in Section 5.4.2, the rescoring experiments demonstrate the usefulness of some variants of BiSLM in choosing better translation hypotheses from a set with precomputed alignments and accordingly projected source parses. Our next step is to integrate the BiSLM as a feature into the decoder (Section 5.4.3). This way we test whether the hypotheses preferred by our BiSLM contain more accurate alignments, and therefore the integrated BiSLM feature will help explore hypotheses with better reordering during translation search.

We run experiments on Arabic-English and Chinese-English. The data setup, the baseline features, training and tuning details are provided in Section 3.5.3. Like before, we use case-insensitive BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2009) and TER (Snover et al., 2006) as evaluation metrics.

The sections are grouped by experiment types, and each contains results for both language pairs. Section 5.4.1 contains baseline and comparison system results, Section 5.4.2 contains rescoring experiments, Section 5.4.3 contains experiments with a decoder-integrated BiSLM.

5.4.1 Baseline and comparison systems

In this subsection we present results of translation experiments for the baseline system and the comparison system, which is the five syntactic-based soft constraints proposed in Cherry (2008) and Bach et al. (2009) (see Section 3 for a detailed description of the constraints). As in the previous chapter, we present the metrics' scores on the concatenated set of all the test benchmarks (Table 5.1 for Arabic-English, Table 5.5 for Chinese-English), as well as on each benchmark individually (Tables 5.2-5.4 for Arabic-English, Tables 5.6-5.8 for Chinese-English).

As can be seen from the tables, the cohesion constraints do not provide additional

Table 5.1: BLEU, METEOR and TER scores for Arabic-English experiments evaluated on a concatenated of all the test benchmarks (MT02, MT03, MT05, MT06, MT08, MT09).

	MT02-MT09 concatenated		
	BLEU	METEOR	TER
PBSMT baseline	51.54	70.82	43.30
cohesion constraints	51.41	70.82	43.33

Table 5.2: BLEU scores for Arabic-English experiments.

	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	64.09	53.46	56.80	46.42	45.84	48.66
cohesion constrains	64.00	53.42	56.77	46.31	45.61	48.49

Table 5.3: METEOR scores for Arabic-English experiments.

	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	75.72	74.27	75.95	67.37	67.24	69.62
cohesion constrains	75.72	74.32	76.0	67.4	67.18	69.57

Table 5.4: TER scores for Arabic-English experiments.

	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	37.45	41.74	38.56	46.20	47.23	43.93
cohesion constrains	37.50	41.64	38.53	46.28	47.33	43.91

Table 5.5: BLEU, METEOR and TER scores for Chinese-English experiments evaluated on a concatenated of all the test benchmarks (MT02, MT03, MT05, MT06, MT08).

	MT02-MT08 concatenated		
	BLEU	METEOR	TER
PBSMT baseline	31.68	59.14	58.76
cohesion constraints	31.62	59.02	59.00

Table 5.6: BLEU scores for Chinese-English PBSMT baseline and BiLM pipelines.

	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	33.31	33.55	33.64	32.59	25.93
cohesion constraints	33.06	33.32	33.49	32.52	25.98

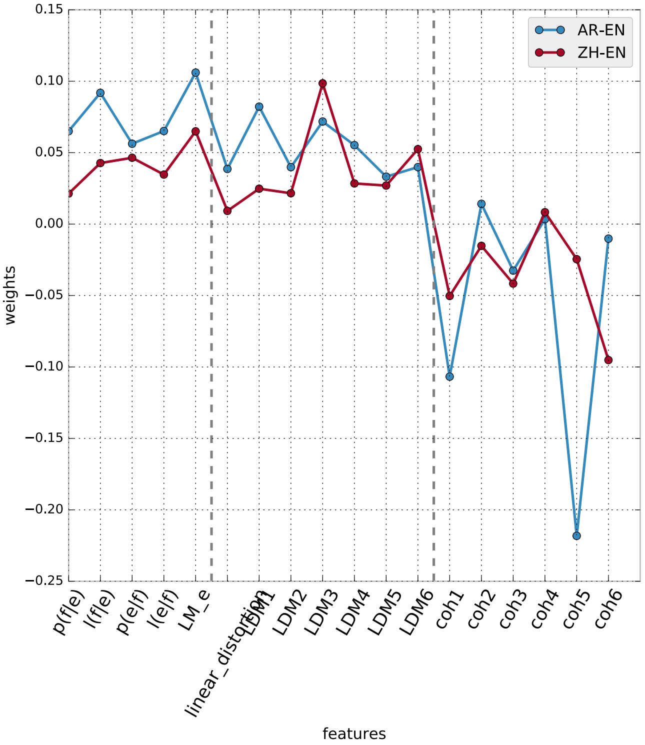


Figure 5.6: Feature weights for systems tuned with baseline features and the six syntactic cohesion features from (Bach et al., 2009). We include a subset of all the features (phrase and word penalty are omitted). The cohesive constraints are the following: **coh1** – simple binary cohesion feature, **coh2** – simple count-based cohesion feature, **coh3** – exhaustive binary cohesion feature, **coh4** – exhaustive count-based cohesion feature, **coh5** – exhaustive count-based noun cohesion feature, **coh6** – exhaustive count-based verb cohesion feature.

improvements over the baseline. The conclusion holds across metrics, test sets and language pairs. This result tells us that simply imposing a syntactic constraint on the translation search does not help find better translation hypotheses. Figure 5.6 shows tuned feature weights for both language pairs. For Arabic-English, the cohesion feature weights are the lowest among all the presented features. For Chinese-English, the exhaustive count-based cohesion constraint (coh4) is approximately at the level of the linear distortion level, but overall the cohesion weights get the lowest values. We also note that the cohesion feature weights for both language pairs are either negative or close to zero.

Our experimental result is different from the one in (Bach et al., 2009), where they also evaluate on Arabic-English and Chinese-English (the size of training data being comparable to ours) and report improvements over the baseline. We first note that

Table 5.7: METEOR scores for Chinese-English PBSMT baseline and BiLM pipelines.

	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	60.71	59.91	61.39	59.58	54.79
cohesion constraints	60.48	59.74	61.32	59.45	54.78

Table 5.8: TER scores for Chinese-English PBSMT baseline and BiLM pipelines.

	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	59.50	58.72	59.09	57.03	60.04
cohesion constraints	59.94	58.98	59.51	57.11	60.17

they evaluate BLEU on a much smaller test set than we do, and, more importantly, their baseline does not contain lexicalized distortion models. As Figure 5.6 suggests, lexicalized distortion models are important features, having weights in the same range as the core models, such as the translation models and target language model.

5.4.2 Rescoring experiments

The purpose of the rescoring experiments is to understand the usefulness of the structured language model for translation when alignments are fixed, and to select a better projection, parsing and labeling algorithm for BiSLM. We consider the following variations of BiSLM:

- whether to use a strong or weak definition of a complete subtree (Section 5.3.2) – **weak/strong completeness**;
- whether to adjoin unaligned target words to a preceding head (Section 5.3.1) – **unalign-adjoin+/-**;
- we compare several target-side labeling methods (Section 5.3.3): plain (just target words), reduce (**LR** or **RR**) or reduce-POS (**LR**_POS or **RR**_POS, where POS is the tag of the root of the reduced subtree).

The rescoring procedure is performed as follows: We extract an n-best list of translation hypotheses produced by the baseline system. We set $n = 1000$. For each translation hypothesis in the list, we extract its derivation sequence from the decoding lattice (sequence of phrase pair applications), and for each phrase pair in the derivation we extract its internal word alignment stored in the phrase table (see Section 5.3.4 for more implementation details). The derivation history and phrase-internal alignments are sufficient to reconstruct the full word alignment between the source and the target hypothesis, and we use it to construct a projected parse of the target side. For the BiSLM variants involving POS-reduction labeling, for each phrase pair in the derivation we extract a sequence of corresponding target POS labels (also stored in the phrase table).

Table 5.9: BLEU, METEOR and TER scores for Arabic-English decoding experiments with BiSLM evaluated on a concatenated of all the test benchmarks (MT02, MT03, MT05, MT06, MT08, MT09). **Statistical significance notation**: improvements are marked ▲ at the $p < .01$ level and △ at the $p < .05$ level.

	MT02-MT09 concatenated		
	BLEU	METEOR	TER
PBSMT baseline	51.54	70.82	43.30
reduce-weak-emptyAdj+	52.06 ▲	71.10 ▲	43.05 ▲

For the projection strategy, we construct its own parse and score the target hypothesis with the corresponding variant of BiSLM. We normalize the resulting BiSLM score by the length of the target hypothesis sequence. The final score used to rank each hypothesis is an interpolation of the original baseline system score and the normalized BiSLM score:

$$\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1 - \lambda) \cdot score_{\text{Baseline}} \tag{5.5}$$

We do not set or select the value of the interpolation weight λ beforehand, but directly run a grid search on the concatenated set of all available test sets. The results are presented in Figures 5.8(a)-5.8(b) for Arabic-English and Figures 5.8-5.9(b) for Chinese-English.

For Arabic-English, we can see that none of the rescoring models outperform the baseline. The overall picture is that the more we rely on BiSLM in rescoring, the worse is the performance. Additionally, we observe that BiSLM variants where the **unalign-adj** has the value **+** perform worse. The **weak/strong** cpmleteness distinction does not appear to play a role. Finally, not applying reduction labeling appears to have a somewhat positive effect. Even though the result of rescoring to Arabic-English is negative, we picked one variant of BiSLM (**unalign-adj +**, weak completeness, reduction labeling) to test in the decoding experiment, since it demonstrated a slight improvement above the baseline for METEOR.

For Chinese-English, rescoring produces statistically significant improvements for BLEU of up to 0.4 BLEU on the concatenated test set. Another finding is that the optimal values are achieved when λ (the interpolation weight) is between 0.2 and 0.4. As for the individual hyperparameters of BiSLM, we again observe that the strong/weak completeness distiction does not play a role. For the **unalign-adj** hyperparameter, on the contrary to Arabic-English, the **-** value produces higher scores. Finally, no reduction labeling (plain) or simple reduction labeling give better results than POS-reduction labeling.

5.4.3 Decoding experiments

In this section we further test BiSLM by fully integrating it into our decoder for tuning and testing. For each language pair, we picked a BiSM system which showed the best results in the rescoring experiments.

For Arabic-English we tested a BiSLM variant with the hyperparameters set as:

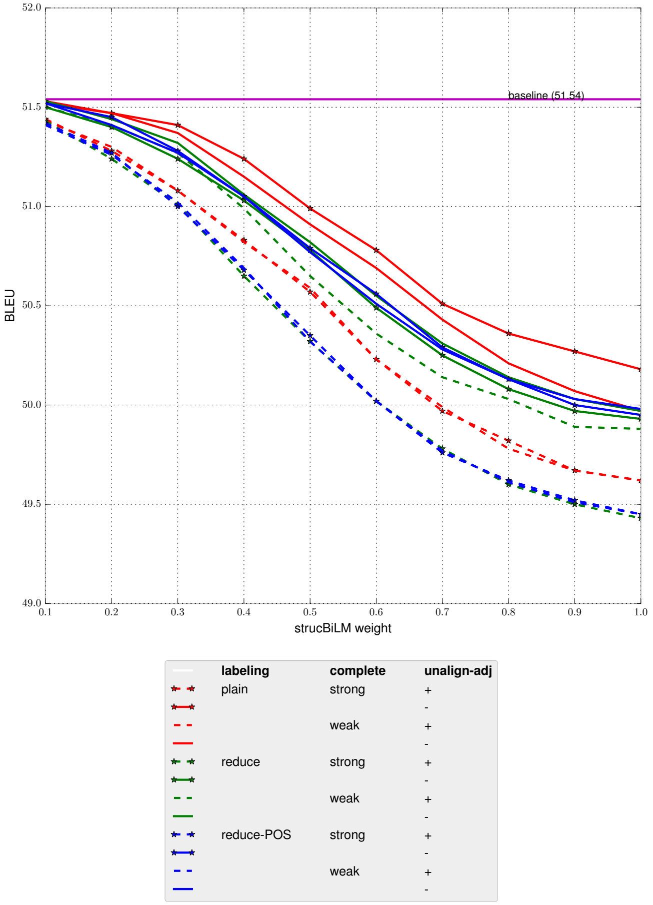
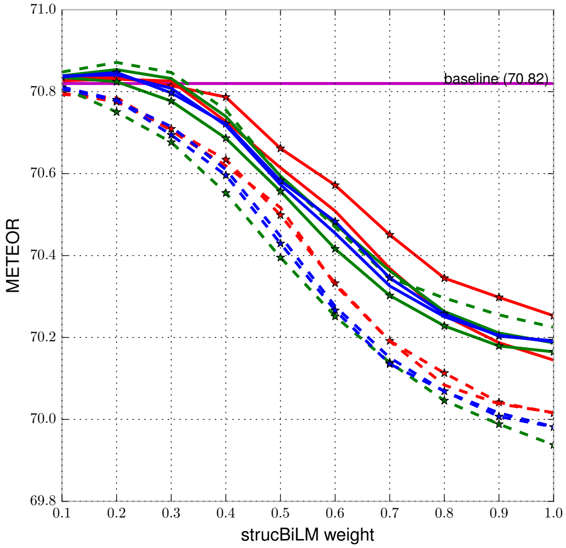


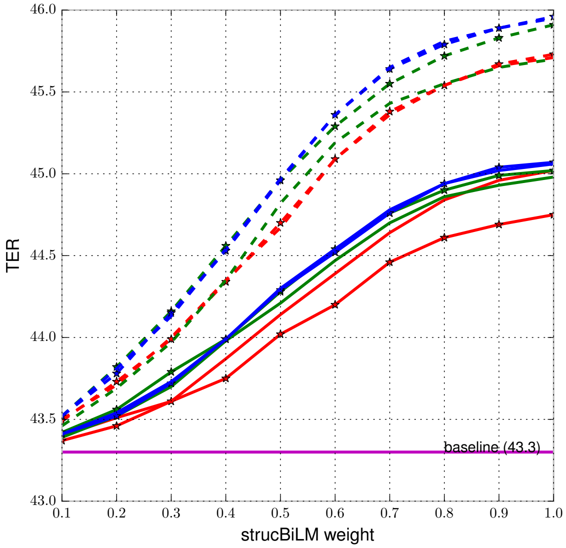
Figure 5.7: BLEU scores for Arabic-English rescoring experiments. The x-axis is the weight λ in the interpolation $\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1-\lambda) \cdot score_{\text{Baseline}}$.

Table 5.10: BLEU scores for Arabic-English decoding experiments with BiSLM. Statistical significance notation is explained in the caption to Table 5.9.

	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	64.09	53.46	56.80	46.42	45.84	48.66
reduce-weak-emptyAdj+	64.90 ▲	54.32 ▲	57.02	46.87 ▲	46.21 ▲	49.39 ▲



(a) METEOR scores for Arabic-English rescoring experiments. The x-axis is the weight λ in the interpolation $\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1-\lambda) \cdot score_{\text{Baseline}}$.



(b) TER scores for Arabic-English rescoring experiments. The x-axis is the weight λ in the interpolation $\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1-\lambda) \cdot score_{\text{Baseline}}$.

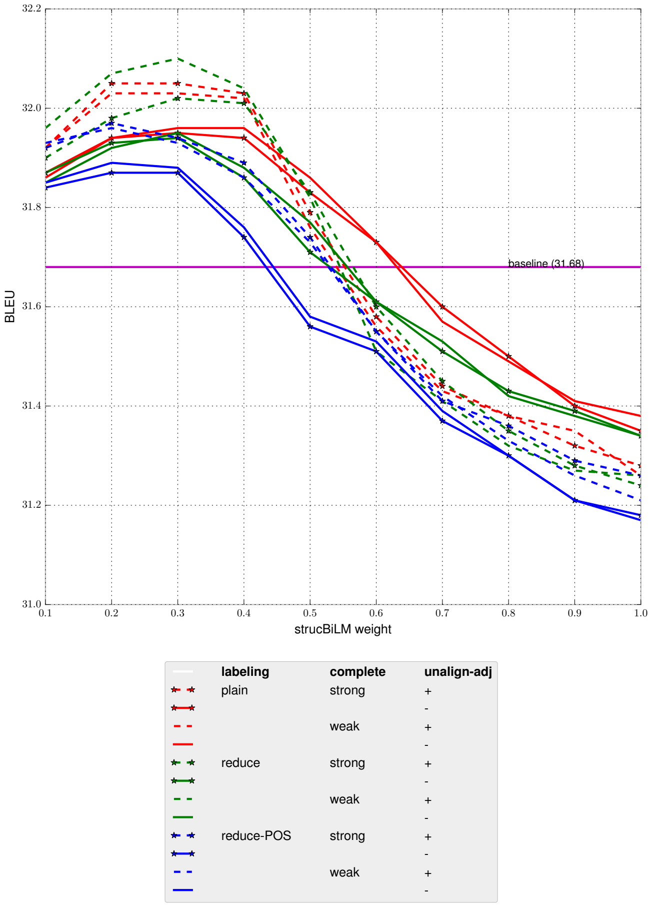
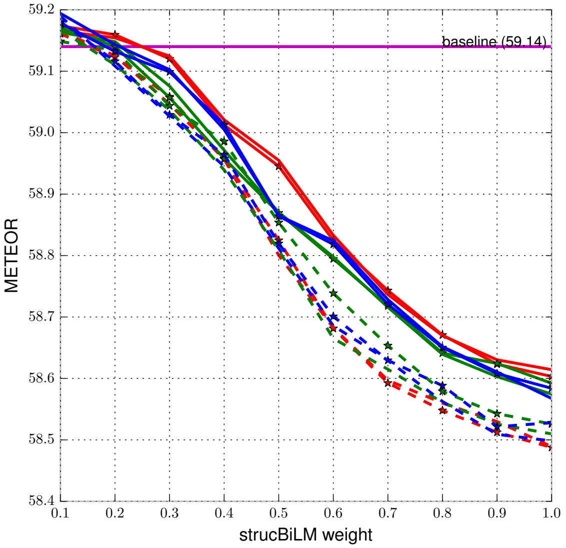


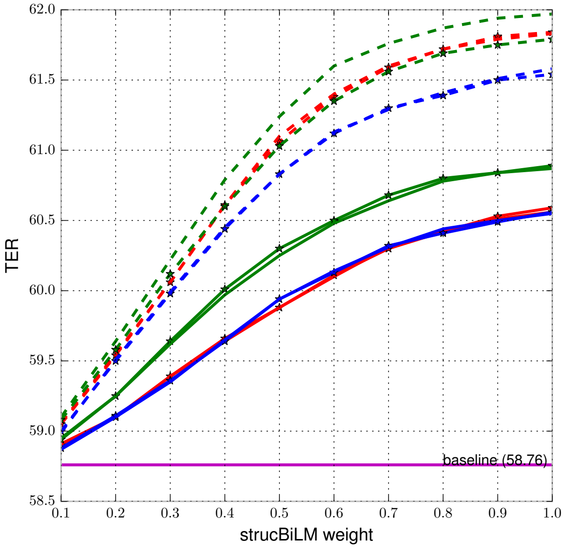
Figure 5.8: BLEU scores for Chinese-English rescoring experiments. The x-axis is the weight λ in the interpolation $\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1-\lambda) \cdot score_{\text{Baseline}}$.

Table 5.11: METEOR scores for Arabic-English decoding experiments with BiSLM. Statistical significance notation is explained in the caption to Table 5.9.

	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	75.72	74.27	75.95	67.37	67.24	69.62
reduce-weak-emptyAdj+	75.94 △	74.54	76.08	67.82 ▲	67.44 △	69.92 ▲



(a) METEOR scores for Chinese-English rescoring experiments. The x-axis is the weight λ in the interpolation $\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1-\lambda) \cdot score_{\text{Baseline}}$



(b) TER scores for Chinese-English rescoring experiments. The x-axis is the weight λ in the interpolation $\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1-\lambda) \cdot score_{\text{Baseline}}$

Table 5.12: TER scores for Arabic-English decoding experiments with BiSLM. Statistical significance notation is explained in the caption to Table 5.9.

	MT02	MT03	MT05	MT06	MT08	MT09
PBSMT baseline	37.45	41.74	38.56	46.20	47.23	43.93
reduce-weak-emptyAdj+	37.01 ▲	41.30 △	38.42	46.04	47.03	43.66 △

Table 5.13: BLEU, METEOR and TER scores for Chinese-English decoding experiments with BiSLM evaluated on a concatenated of all the test benchmarks (MT02, MT03, MT05, MT06, MT08). Statistical significance notation is explained in the caption to Table 5.9.

	MT02-MT08 concatenated		
	BLEU	METEOR	TER
PBSMT baseline	31.68	59.14	58.76
reduce-weak-emptyAdj-	31.97 ▲	59.22	58.70

Table 5.14: BLEU scores for Chinese-English decoding experiments with BiSLM. Statistical significance notation is explained in the caption to Table 5.9.

	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	33.31	33.55	33.64	32.59	25.93
reduce-weak-emptyAdj-	33.95 ▲	33.81	33.81	32.79	26.15

Table 5.15: METEOR scores for Chinese-English decoding experiments with BiSLM.

	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	60.71	59.91	61.39	59.58	54.79
reduce-weak-emptyAdj-	60.90	59.79	61.56	59.77	54.76

Table 5.16: TER scores for Chinese-English decoding experiments with BiSLM.

	MT02	MT03	MT05	MT06	MT08
PBSMT baseline	59.50	58.72	59.09	57.03	60.04
reduce-weak-emptyAdj-	59.29	58.74	58.96	57.00	60.03

unalign-adj +, weak completeness, reduction labeling. The results are presented in Tables 5.9-5.12 on a concatenated test set and individual test sets. Unlike for the rescoring experiments, we observe significant improvements for all metrics and across almost all test sets. This result is different from rescoring experiments which did not demonstrate any significant improvements. It suggests that the alignments produced by the baseline are quite different from those learnt by the BiSLM.

For Chinese-English we tested a BiSLM with hyperparameters set to **unalign-adj -**, weak completeness, reduction labeling. The results are presented in Tables 5.13-5.16. One can see that statistically significant improvements are only achieved for BLEU. Moreover, the improvements are lower than for rescoring.

It is interesting to observe that for the two language pairs the results for rescoring and the decoding experiments demonstrate opposite patterns. It entails that the relationship between a high BiSLM score and good translation quality is not straightforward. Our hypothesis was that if we have a good language model that recognizes systematic structural correspondences between source and target sentences, then it could serve as a feature during decoding that steers the translation process towards hypotheses which conform to this systematic correspondence. However, for Chinese-English this appears not to be the case completely. A possible explanation is that the empirically correct correspondences are only a subset of those ranked highly by BiSLMs. For Chinese-English, the n-best list could have just contained the translation hypotheses with correct correspondences, and the model could pick them up. On the contrary for Arabic-English, it could be that the n-best only contained correspondences that are incorrect but ranked highly by BiSLM. We leave the investigation of this question to future work. One possible direction is trying to control the expressivity of BiSLM by adding more hyperparameters, in addition to the three currently used parameters.

5.5 Conclusions

In this chapter we proposed a novel way to adapt structured language models to phrase-based SMT. Our method requires minimal changes to the PBSMT pipeline. Our method is based on the idea that there exists some systematic correspondence between source and target sentential structures, which can be induced by a combination of elementary projection rules from the source structure onto the target side. We contrast our method to a constraint-based approach to cross-lingual syntactic correspondence, the implementation of which did not result in translation quality improvements. We tried a number of variations of our model and evaluated them in rescoring experiments, resulting in statistically significant improvements for Chinese-English. Based on the rescoring experiments, we picked the best performing model variants for Arabic-English and Chinese-English and evaluated them in the experiments where the BiSLM was fully integrated into the decoder. We observed improvements for both language pairs. We observed that for Arabic-English translation quality improves in decoding experiments but not rescoring ones. For Chinese-English, we saw improvements in both experiments, however, for rescoring they were greater. We hypothesized that this is due to the fact that BiSLMs in their current form are too expressive and may upvote hypotheses incorporating incorrect structural correspondence.

Finally, we return to the research questions posed at the beginning of the chapter:

RQ2 Is there a systematic mapping between source and target syntactic representations in a parallel sentence and can it be used to improve translation?

This is a fundamental question and we answer it (partially) by addressing the subquestions below.

RQ2.a. Is there a universal characterization of a mapping between source and target structure? Can this characterization be used to constrain the decoding process to produce better translations?

In order to answer this question, we implemented and tested a model realizing such an approach, namely the syntactic cohesion model. Including it in form of soft constraint features did not result in an improvement of translation quality for either of the language pairs.

RQ2.b. Can the mapping be defined in terms of projection constraints between *elementary* parts of source and target structures? Can we fit a statistical model over the resulting corresponding source and target structures to characterize the overall mapping?

We proposed a simple set of rules, which projects a source structure onto the target. We used a structured language model to learn a characterization of target sequences parsed in this fashion. The model yielded statistically improvements for both Arabic-English and Chinese-English in rescoring and decoding settings. This result suggests that the BiSLM indeed captures the systematic relation between source and target syntactic structures. This correspondence can be used in a feature to improve translation quality, as suggested by improvements in translation metrics. At the same time, we have some evidence that the models in their current form are too expressive and also accept incorrect translation pairs.

RQ2.c. What are the important mapping constraints that result in structured language models improving translation output?

We found that the parameter regulating how to incorporate unaligned target words had the most effect. The parameter regulating how to deal with non-cohesive translation did not appear to have an effect, suggesting that translations (for our given baselines) actually tend to be cohesive. This is also a potential explanation for why the cohesive constraints from **RQ2.a** did not contribute to translation quality. Finally, we experiments with a few labeling strategies and found that no or very simple labeling produced better metric scores. Decorating exposed head with POS information of their children resulted in worse performance.

Syntactic correspondence between source and target sentences is a fundamental question in SMT. The answers to the research questions we provided here are not exhaustive and suggest a number of interesting future research directions to extend the proposed model.

The original paper underlying this chapter was written shortly before neural machine translation became the focus of academic research in MT. Writing this chapter now, we ask ourselves: can the discussed ideas and models be transferred to NMT? First of all, the DCA is formulated using the notion of word alignment, and the concrete models of correspondence (such as syntactic cohesion or our model) are grounded in the word alignment computed at the beginning of the SMT pipeline. State-of-the-art NMT (Bahdanau et al., 2015) is not grounded in word alignments, but has an internal mechanism approximating them (attention). However, while for SMT alignment is the central concept, for NMT it is secondary. In the light of this consideration, it is not entirely straightforward to translate DCA from SMT to NMT from a conceptual point of view. On the other hand, if we disregard this consideration, and assume that alignments are the word alignments of NMT, then the models discussed in this chapter can be reinterpreted for NMT. In fact, Chen et al. (2017) have very recently proposed a model of a NMT decoder augmented with a source coverage mechanism[8] (Tu et al., 2016) which computes a coverage representation for each source word conditioned on the hidden state of the decoder at step i, as well as the attention and coverage representations of the word's left-hand and right-hand children in a dependency tree. The attention representation is computed by re-using the coverage representation from the previous step. Conceptually, this model is designed to condition an attention weight of a word based on how much it and its dominated source subtree were attended before. This is directly related to the idea of syntactic cohesion.

As for our BiSLM, there are a number of conceivable ways in which it could be integrated into NMT. First of all, it can be used as an additional model,[9] either at a rescoring stage or at a decoding stage (Gulcehre et al., 2017). A more interesting way would be to implicitly incorporate the parsing procedure of the BiSLM into the decoder. This could be done for instance by adding a hidden state h_i^{struc} which is a function of this state at the previous decoding step h_{i-1}^{struc} and the product of the current attention distribution and some source word representations which incorporates dependency information (for example, concatenation of a word's vector and its parent's vector, or a graph convolution representation centered around the given word, as in (Bastings et al., 2017)).

[8] Inspired by the coverage vector of the phrase-based SMT.

[9] In this case one has to convert real-valued attentions into binary alignments

Part II

Exploring Diversity in Neural Machine Translation

6
Background: Ensembles, System Combinations, and Baselines

In this chapter we provide relevant background for Chapter 7, where we propose and compare methods of multi-source ensemble combination in neural machine translation (NMT). As we explain in Chapter 7, the sequence-to-sequence model of translation is simple and generic enough to directly apply the general ensembling methods from machine learning, which we review in Section 6.1. In Section 6.2 we give some background on previous research on system combination in the context of statistical machine translation (SMT). Finally, in Section 6.3 we report the details of how we build the NMT system that we use in the experiments in Chapter 7.

6.1 Ensembles in machine learning

In this section we introduce the basic concepts and main algorithms used for ensemble combination of machine learning systems. We concentrate on ensemble methods for classification, since this is the focus of the research in the next chapter.

The general idea behind ensemble combination is to aggregate predictions (given the same input) of various *diverse* prediction systems (Brown et al., 2005). On an intuitive level, such an approach is likely to improve the overall prediction quality because in a set of diverse predictors, some of them will make correct predictions on some subset of inputs and make errors on the other ones, on which some other predictors are likely to give a correct answer. On a somewhat more formal level, if for every input we aggregate predictions in a reasonable way and under the assumption that the ensemble set is diverse enough so that the errors that the individual predictors make are uncorrelated, then on average (over all input instances) the errors of individual predictors will cancel each other out due to the fact that other predictors will make correct and confident predictions.

The discussion above entails that two things are important to obtain a well-performing ensemble system: First, an ensemble set should consist of diverse predictors whose distributions of errors are uncorrelated, see (Kuncheva and Whitaker, 2003) for a study of measures of ensemble diversity. Second, the combination method should be able to recognize the strengths and weaknesses of individual predictors to combine them optimally. The two components can be determined independently when building an ensemble system, which we do in Chapter 7. Some algorithms also provide a method to jointly create a diverse ensemble set optimized towards a particular task and combination method. Rokach (2010) terms the two approaches as *independent* and *dependent*, respectively. For example, Boosting (Freund et al., 1996) iteratively trains a set of weak learners, where at each iteration a new weak learner is trained on a subset of training data on which the previously trained weak learners made more errors.

There are a few common strategies of independently inducing a diverse ensemble set (Brown et al., 2005). One is to create a set of separate training sets from one common training set and train separate classifiers on them. For example, Bagging (Breiman, 1996) obtains new datasets by sampling. Another method is to use different learning algorithms on the same dataset. Finally, for learning methods depending on an initial parameter setting, different initializations can induce diverse predictors.

Combination methods vary with respect to how much information they use to combine the predictions and exactly what they combine. Perhaps the simplest way to combine predictions is by doing majority voting over the predicted labels. A more sophisticated way is to actually take the class probability distributions into account. Uniform averaging of class distributions is a common method, and we use it as a baseline in Chapter 7. Another method is defining a so-called belief function for each class separately based on the predictors' distributions (Shilen, 1990). These kinds of methods do not take individual strengths and weaknesses of predictors into account. A combination method can also be trainable, such that it learns the systems' prediction behavior on a held-out data set. *Stacking* (Wolpert, 1992) is a method of meta-classification: a meta-classifier is trained to take the outputs of individual predictors to output the final prediction. *Mixture of experts* (Jacobs et al., 1991) is a method of contextual combination: based on the given input, which is common to all the predictors, a trainable mixture gating network decides what weights to assign to each individual prediction and then aggregates them by (weighted) summing.

6.2 System combination in machine translation

In this section we provide an overview of previous work on system combination in machine translation. Here we focus on previous research in SMT (and specifically, phrase-based SMT) and address system combination of NMT systems (ensembling) in the next chapter.

Combination approaches in phrase-based SMT require insights beyond the kind of methods that we described in Section 6.1 on ensembles in machine learning. This is because it is not easy to factorize the prediction $p(E|F)$ of E in phrases into smaller components, as different features in the log-linear combination, see Equation 2.3 in Section 2.1, capture a distribution over different kinds of units (phrases, words). More importantly, during prediction the full distribution over the next step in a derivation (Equation 2.6 in Section 2.1.2) is not available due to the intractability of the resulting search. Therefore, the methods from the previous section cannot be applied in a straightforward manner.

Another type of approach proposes a way to combine predictions produced from different inputs: Och and Ney (2001) perform combination at the level of complete translation hypotheses, obtained by feeding a different input to the same pre-trained translation system. They approximate the distribution over the whole set of translations by a set of 1-best translations, one for each input sentence. The final translation hypothesis E^* is selected based on a formula involving an aggregate score of $p(F_1|E^*), ..., (F_N|E^*)$, for every input sentence F_i.

Another line of research (Bangalore et al., 2001; Matusov et al., 2006; Rosti et al., 2007) studies methods to recombine different translation hypotheses at a sub-sentential level. This is achieved by constructing a confusion network of sub-sentential units from the hypotheses and then choosing a path in the network via consensus translation.

Finally, Schroeder et al. (2009) combine different inputs prior to translation into an input lattice (Xu et al., 2005; Dyer, 2009). The challenge of this approach is to combine input sentences in different languages into one lattice structure. This approach is motivated by the fact that different languages exhibit different word orders and orders in lattices should be transitive. Thus this method is only applicable to languages with approximately similar word orders.

6.3 NMT baseline and experimental setup

In this section we provide details on our NMT baseline. The general background on NMT can be found in Section 2.2. Here we only report the choice of concrete neural models and hyper-parameter settings and give details on the training and decoding procedures (Section 6.3.3). We also describe the data selection (Section 6.3.1) and preprocessing procedures (Section 6.3.2), including vocabulary selection, which is necessary to define the output layer of the NMT system.

6.3.1 Data

In the next chapter we describe a method to combine predictions of NMT systems for different language pairs with a common target language, which is also known as a multi-source translation scenario. For this, we choose to experiment with German-English and French-English. We choose these languages to introduce diversity into ensembles (see Section 6.1 on the importance of diversity). German is structurally substantially different from both English and French, while English and French are structurally similar and are typically easily mutually translatable (Bojar et al., 2014). We expect a French-English system to perform better than a German-English system. But also, given the linguistic intuition about the structural differences between these translation pairs, we hope that the two systems compensate for each other's weaknesses when combined in an ensemble.

To ensure that the only distinguishing factor between different language pairs is the source language, we choose training data that is to a large extent parallel across all three languages, i.e., it is a trilingual parallel text with small bilingual portions. To this end, we train all of our systems on the TED talks data set (Cettolo et al., 2012). All available data is split into a training and a validation set to train individual NMT systems, a training and a validation set to train a combination function for ensembles, and a separate test set for the final evaluation. The training data for learning the ensemble combination function and the test set must be fully parallel across all languages (tri-parallel), see details in Section 7.3. Therefore, we extract our test set from the available trilingual data and do not use the test sets provided by Cettolo et al. (2012) since they are not parallel across all three languages. Of course, the test set does not overlap with the training data. Table 6.1 provides some statistics of the data.

6.3.2 Data preprocessing

Prior to splitting the data into the training, validation, combination training, and testing data sets, we perform some basic data preprocessing: tokenization (with a simple in-house tokenizer), lowercasing, and deduplication.

In NMT, it is important to predefine the source and target vocabularies in advance, since the former defines the dimensions of the source embedding tensor, and the latter defines the dimensions of the target embedding tensor and the dimensions of the output layer. This is usually done by sorting observed word types based on their frequency and selecting the top k, while mapping the rest of the word types to $\langle \textit{unk} \rangle$ (Sutskever et al., 2014). For the French and German source sides, we select the top 35,000 words. For the target side, it is necessary to make the output vocabularies exactly the same for both language pairs at the ensemble combination stage. To this end, we precompute the intersection of the target vocabularies of the two language pairs and rank the word types by their summed frequencies in order to select the top k target words. We set k to 24,000 for English, i.e., the target language.

6.3.3 NMT system: model details, training, decoding

Our baseline is a sequence-to-sequence model with attention, see Section 2.2.1. We use the global attention mechanism with the *dot score* function from (Luong et al., 2015a):

$$\textit{score}(h^t, H^s) = (h^t)^\top h^s, \tag{6.1}$$

where *score* is a distribution vector over source states which is fed to a *softmax* function (see Equation 2.16 in Section 2.2.1).

The source encoder is a four-layer unidirectional LSTM. The final hidden states of the encoder are used to initialize the decoder, which is also a four-layer unidirectional LSTM. We set the size of all embeddings and hidden layers to $1,000$. We use LSTM units for the recurrent hidden states and apply dropout with a probability of 0.2 (Srivastava et al., 2014).

For network training we use the in-house NMT system Tardis implemented in Torch.[1] All parameters are initialized by randomly drawing from a uniform distribution, except for the embeddings, which are initialized by sampling from a Gaussian with unit variance. We optimize the network with respect to the negative log-likelihood, see Equation 2.17 in Section 2.2.2. We use stochastic gradient descent with mini-batches of size 20 with a learning rate of 1 and a decay rate of 0.8 after the fifth epoch. Each translation system is trained for 20 epochs. During training we limit the lengths of

[1] https://github.com/ketranm/tardis

Table 6.1: Preprocessed data statistics for (a) NMT training, (b) ensemble combination function training, and (c) testing.

	Set	N. of lines	N. of word tokens	N. of word types
(a)	train DE-EN	123,955	DE: 2,3M; EN: 2,5M	DE: 89K; EN: 41K
	train FR-EN	127,755	FR: 2,9M; EN: 2,6M	FR: 58K; EN: 41,9K
	valid DE-EN	2,052	DE: 40.3K; EN: 41.5K	DE: 6.3K; EN: 4.7K
	valid FR-EN	887	FR: 21.5K; EN: 20K	FR: 3.7K; EN: 3.1K
(b)	combination train DE-EN-FR	19,000	DE: 372K; FR: 447K; EN: 396K	DE: 29K; FR: 22.8K; EN: 17K
	combination valid DE-EN-FR	1,000	DE: 19K; FR: 23K; EN: 20.5K	DE: 4.3K; FR: 4K; EN: 3.5K
(c)	test DE-EN-FR	3,000	DE: 62.9K; FR: 78K; EN: 69.5K	DE: 9.3K; FR: 8.3K; EN: 6.5K

predicted sequences to 50 tokens. For each language pair we train four systems by sampling different initial parameter values.

Decoding is done with beam search. In all of the translation experiments the beam decoding size is set to 20. We evaluate performance with BLEU Papineni et al. (2002) and METEOR Lavie and Denkowski (2009), see Section 2.3.1 for a description of the metrics.

7 Ensemble Learning for Multi-Source Neural Machine Translation

7.1 Introduction

It has been shown for various machine learning applications that combining multiple systems, referred to as *ensembling*, can substantially improve performance (Wolpert, 1992; Dietterich, 1999; Kuncheva and Whitaker, 2003; Rokach, 2010). In this chapter we explore ensemble combinations of neural machine translation systems at decoding time.

System combination has also been successfully applied to statistical machine translation system (SMT) (Och and Ney, 2001; Matusov et al., 2006; Schwartz, 2008; Schroeder et al., 2009). However, system combination methods in the phrase-based (Koehn et al., 2003, PBSMT) and hierarchical (HSMT) frameworks (Chiang, 2007) tend to be rather complex, requiring potentially non-trivial mappings between partial hypotheses across the search spaces of the individual systems. For this reason SMT system combination is often limited to combining hypotheses from the n-best lists (Och and Ney, 2001). Alternatively, SMT systems can also be combined by combining different inputs in a single structure as is the case for multilingual system combination (Matusov et al., 2006; Schroeder et al., 2009). Unfortunately, input sentences in different languages may have very different structure, requiring elaborate methods to align sentences, which means that multilingual system combination is in practice restricted to languages with similar structures.

In contrast, the recently emerged neural machine translation (NMT) framework offers a straightforward way to combine multiple systems. Most of the current NMT architectures (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015) formalize target sentence generation as a word sequence prediction task. At each time step during sequence prediction, a translation system outputs a *full* probability distribution over the target vocabulary. Therefore, the task of NMT system combination can be cast as an ensemble prediction task and a variety of existing general prediction combination methods can be applied. While this chapter focuses on word-based models, the ensemble methods discussed in this chapter can be applied to character-based sequential NMT models (Ling et al., 2015) in a very similar fashion.

Ensemble prediction is commonly used in NMT. It is typically used during the decoding stage, where multiple pre-trained NMT systems are combined to make a prediction. A commonly reported method is uniform weighting of the output layers, i.e., distributions over the target vocabulary, produced by different trained instances of the same NMT architecture for the same language pair (Bojar et al., 2014). We use this method as a baseline where variations of the same system are produced by different parameter initializations. Alternatively, it is also possible to take parameter snapshots from different training epochs (Sennrich and Haddow, 2016). Recently, a new type of ensemble has been introduced in NMT: *multi-source ensembles* (Firat et al., 2016), which is a set of NMT systems that translate from different source languages into the same target language. To use this ensemble method, it is necessary to have a multi-parallel input set, whereby input parallel sentences in different languages have the same meaning.

For any method of inducing an ensemble set, the essential criterion is to make the set diverse (Kuncheva and Whitaker, 2003). Intuitively, this is because different predictors are likely to produce slightly different errors for different input instances, and if their predictions are combined, the overall error is reduced. Different random initializations force the same training algorithm to converge to different local optima. Different source sentences may differ quite significantly in their structure and thus present a different training task to an NMT learning algorithm. Multi-source ensembles offer a *linguistic* source of variation for translation systems, which may range from the way particular words are translated to the way the whole sentence is structured. This is in line with the common observation that translation systems trained on language pairs with different source languages differ in their performance (Bojar et al., 2014).

Besides the linguistic interest, multi-source translation can be applied in practical real-life scenarios. Examples include multi-lingual websites, where some content has already been made available in a couple of languages (by human translators) but needs to be further translated into other languages. Another example are parliamentary proceedings, typically available in many languages. Furthermore, Firat et al. (2016) study neural multi-source translation in the context of zero-resource translation, which is a setting where no parallel resources are available for training. In their experiments they show that multi-source pivot-based translation improves translation quality compared to simple pivot-based translation.

In this chapter, we compare ensembles for a number of combination methods and evaluate how much performance gain they provide in the multi-source translation setting. Specifically, we aim to tackle the following research questions:

RQ3 Can we exploit the variation in cross-lingual correspondence and improve translation quality with multi-source NMT ensembles?

RQ3.a. To what extent does ensemble performance depend on the quality of individual translation systems that are part of the ensemble?

RQ3.b. Is there systematicity in what a translation system for a given language pair is good at and bad at? Can we exploit this systematicity for multi-source translation?

RQ3.c. How do multi-source ensembles compare to ensembles of NMT systems for single language pairs trained with different initialization seeds?

All previous approaches employing NMT ensembles do so by applying a simple linear, uniform weighting of the output probability distributions. However, it seems intuitive to assign different weights to different systems, especially in the case of multi-source translation. Also, so far, multi-source ensembles (Firat et al., 2016) have only been evaluated for French-English and Spanish-English. All of the three languages, especially French and Spanish, are very similar, and for this scenario their contribution may be close to equal. In order to explore the diversity offered by linguistic variation, we chose to evaluate on English, German and French. German is structurally substantially different from both English and French, while English and French are quite similar and are typically easily mutually translatable (Bojar et al., 2014).

The inclusion of language pairs of different degrees of translation difficulty also allows to test how much translational difficulty influences ensemble quality (**RQ3.a**). Previous literature already contains a partial answer to this question. Namely, different parameters snapshots of a model's training are models of different degrees of generalization (Sennrich and Haddow, 2016) and thus of different degrees of potential translation quality. In our experiments, however, we are exploring how systems trained to achieve their highest generalization work together.

As we mentioned above, the performance of translation systems can vary with respect to different aspects of translation quality, such as correct reordering or lexical translation choice. In order to combine the strengths of individual systems, we use a held-out set to train different combination functions sensitive to specific prediction contexts. This will help us answer **RQ3.b**. We consider two types of combination: global (fixed weight for every prediction instance) and context-dependent, where weights are estimated for every prediction step. The latter is more fine-grained and is in principle able to capture more linguistic nuance, but may be difficult to train due to data sparsity.

The contributions of this chapter can be summarized as follows:

1. We start with a simple experiment motivating our interest in multi-source ensembles and offering preliminary answers to **RQ3** and its subquestions (Section 7.2).
2. We propose two learning methods of combination function learning (Section 7.3). The combination function is designed to be learnt on a small amount of data. In total we compare three kinds of ensemble combination:
   (a) *uniform* combination (baseline);
   (b) *global* combination: we learn a vector of weights to produce a weighted sum of decoders' outputs during decoding;
   (c) *context-dependent* combination: we train a function that outputs combination weights for each prediction step i during decoding.
3. We evaluate uniform, global, and context-dependent combination methods for two kinds of ensemble induction methods (Section 7.4):
   (a) *monolingual* ensembles, obtained by different random initializations of NMT parameter values;
   (b) *multi-source* ensembles, obtained by using semantically equivalent source sentences in different languages to translate into the same target language

(translation systems with different source languages but the same target language).

Additionally, we propose a hierarchical combination method which prevents overfitting and further improves translation quality.

Since we are in the realm of decoding-time ensembling, we do not modify the internals of an NMT system and it can be viewed as a black box. The details of the NMT architecture, as well as the hyper-parameter settings and training schedules, are provided in the background chapter (Section 6.3). All models in this chapter were trained on TED talks data. As we are working with decoding-time multi-source ensembles, where the source sides have to be parallel, we needed to have a multi-parallel test set. The details of the data setup are provided in Section 6.3.1. We trained German-English and French-English NMT systems. We obtained the test set by sampling 3,000 multi-parallel sentences from the training set (see Section 6.3 details). All translation outputs are evaluated with respect to general purpose translation quality metrics, namely BLEU and METEOR (see Section 2.3.1 for details about the metrics).

7.2 Decoding-time ensemble prediction in NMT and multi-source ensembles

In this section we provide the technical details about how we apply ensembling at decoding time. Additionally, we motivate multi-source ensembles by designing an experiment showing that this kind of ensemble opens up possibilities for non-trivial combination methods (Section 7.2.2).

Generally speaking, in order to specify an ensemble method, one needs to specify how a set of predictors is induced and how they are subsequently combined to make joint predictions (Rokach, 2010). For the construction of the set of predictors, it is essential that they make diverse predictions in order to decrease the prediction error. Our goal in this chapter is two-fold: First, we want to understand how different ensemble induction methods influence the quality of translation. We consider ensembles obtained by different random initializations of the same model prior to training (we refer to them as monolingual ensembles) and multi-source ensembles. Second, we want to find the optimal way to combine individual predictors into an ensemble. In the remainder of this section we provide a detailed technical description of how we do ensemble combination, and present experimental analysis of achievable translation improvements for different NMT ensemble methods.

7.2.1 NMT ensemble combination during decoding

As mentioned above, we concentrate on ensembles at decoding time and do not consider co-training of predictors in an ensemble set. In NMT, the decision of which word to predict is based on the output layer. A standard way to achieve decoding-time ensembling is to combine the output layers[1] of individual translators to obtain an

[1] See definition of output layer in Equation 2.13, Section 6.3.

ensemble prediction (Equation 7.1). The word thus predicted by an ensemble is then fed as input at the next prediction step in a sequence-to-sequence model. Figure 7.1 provides a graphical illustration of NMT ensemble prediction.

$$y^{\mathcal{E}} = \mathit{comb}(y^1, ..., y^m) \tag{7.1}$$

where $y^1, ..., y^m$ are output layers of individual NMT systems in an ensemble.

We would like our method to be applicable to situations where a trilingual parallel corpus—needed to train a multi-source combination function—is a scarce resource, which is a realistic assumption. Therefore we are interested in combination functions with a small number of trainable parameters. In our approach, we concentrate on *scalar* prediction combination:

$$\mathit{comb}(y^1, ...;, y^m) = \sum_{i=1}^{m} w_i y^i, \tag{7.2}$$

where $\sum_i w_i = 1$ are scalar weights.[2]

7.2.2 Exploring combination weights for NMT ensembles

Both single-source ensembles with different initializations Sutskever et al. (2014) and multi-source ensembles have been used before Firat et al. (2016). However all of the previous approaches use simple, uniform weighting. We refer to this method as *uniform combination*, as it does not assume anything about the contributions of the individual predictors. In order to explore the bounds of achievable performance, we perform grid search over the *global* combination weights $\langle w_1, w_2 \rangle$ for a two-element ensemble (i.e., the weights are constant throughout the decoding run). We run the experiment for both monolingual and multi-source ensembles.

The results of the grid search experiments are presented in Figure 7.2 and summarized in Table 7.1. We observe an increase in performance for both BLEU and METEOR for all ensembles. Moreover, we see that the metric scores are higher in the region of 0.5, which justifies uniform ensemble combination. At the same time, none of the graphs are completely symmetric: the highest scores are achieved with a weight value of 0.6 or 0.7 assigned to the stronger system in an ensemble.

It is not surprising that different individual systems may have different relative contributions to the ensemble, and it suggests to further investigate combination methods that could distinguish between the relative contributions of the individual members of an ensemble, as addressed by **RQ3.b**. We will distinguish between two kinds of trainable combination functions: global and context-dependent. The former combines NMT predictions in the same way for every input at every decoding time step. The latter can combine predictions differently depending on the current context during decoding. We describe the corresponding learning methods in more detail in Section 7.3.

[2] In addition to the methods described in this chapter we also considered combining the *logit* layers, i.e., y prior to softmax operation. This can be seen as doing geometric mean combination. However, the resulting ensemble system under-performed the stronger individual system of the ensemble, therefore in the rest of the experiments we proceeded with the arithmetic mean function only.

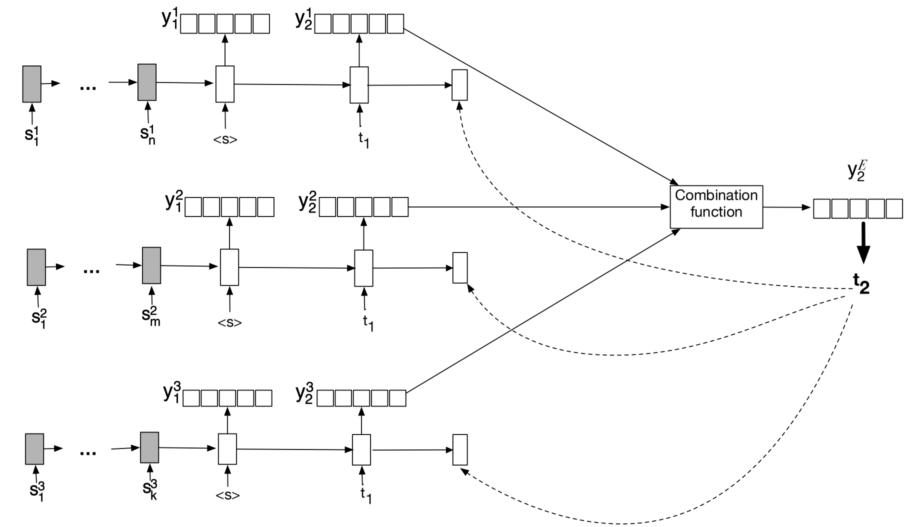


Figure 7.1: Illustration of decoding-time NMT ensemble combination. s_i^j denotes the ith input source word for the jth NMT system (in case of monolingual NMT ensembles $s_i^j = s_i^k$ for all j, k. t_i denotes $i - th$ input target word. For all individual NMT systems, t_i is the same. At each prediction step, each jth individual NMT system produces an output layer y_i^j. Each y_i^k, where k runs over invididual systems' indices is fed to a combination function to produce ensemble output layer $y_i^{\mathcal{E}}$, based on which the next target word t_i is predicted. This target word is fed as input to each of the individual systems at the next prediction step.

Table 7.1: Summary of the grid search of the scalar combination weights depicted in Figure 7.2.

	best system in ensemble		uniform combination		best combination	
ensemble	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
fr_1,fr_2→en	**27.8**	**55.8**	29.2	58.0	29.3	58.1
de_1,de_2→en	20.5	49.10	21.8	50.4	21.9	50.4
fr,de→en	**27.8**	**55.8**	**29.5**	**58.3**	**30.2**	**58.9**

The second major finding of our parameter sweep is that the multi-source ensemble gives a higher upper bound performance than single-source ensembles, even though one of the ensemble members is substantially weaker in its individual performance. This finding reinforces the original linguistic motivation for multi-source ensembles with which we can obtain improvements of up to 0.87 BLEU and 0.77 METEOR over the highest single-source ensemble result. This finding gives a preliminary answer to **RQ3.a**, tackling the relation between individiual quality and ensemble quality, and **RQ3.c**, addressing the differences between monolingual and multi-source ensembles.

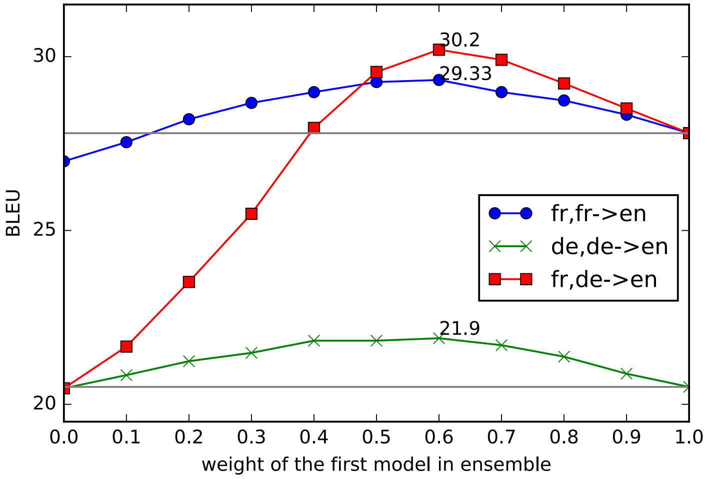
In the following sections we describe how we make use of the uncovered potential of the two types of ensembles. We are especially interested in making full use of the complementary strengths of systems with different source languages.

7.3 Combination function learning

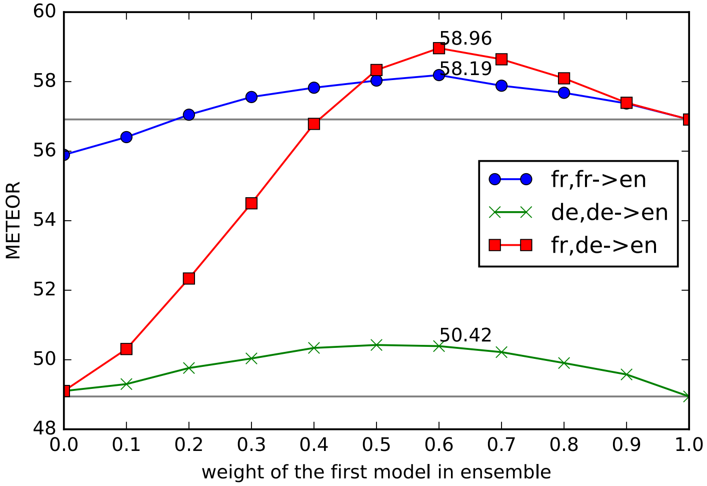
Having established that individual systems do not contribute equally towards correct predictions in single-source and multi-source ensembles, we develop an approach that is capable of training a function that can combine them optimally. For the case of multi-source ensembles the combination training set is a trilingual set consisting of $19,000$ lines (see Section 6.3.1 for details). We deliberately chose a small data set to establish how applicable the method is in a low-resource scenario. We also use this training set to train single-source ensembles. In this section we present two kinds of combination models, as well as their training details.

First, when a scalar combination vector is fixed for every prediction step, we refer to it as *global combination*. The optimized function is a vector $\langle w_1, ..., w_m \rangle$, where m is the size of the ensemble set. We train it with AdaGrad (Duchi et al., 2011) for 10 epochs with a learning rate of 0.001.

Second, we explore a more fine-grained combination method, where the contributions of individual predictors are assessed based on the decoding state. We adapt a mixture of experts model (Jacobs et al., 1991) to learn the *context-dependent combination*. The original mixture of experts model works as follows: We have a set of experts (predictors) and an input x, which is fed to each of the predictors. x is also fed to a gating network which outputs weights for each of the experts. The resulting prediction is a weighted sum of experts:



(a) BLEU scores for Fr, De into En ensembles.



(b) METEOR scores for Fr, De into En ensembles.

Figure 7.2: Results of a 2-ensemble parameter sweep for the two types of ensemble induction. The x-axis represents the value of the first combination weight w_1. Number-marked points are the maximal observed scores for a given ensemble. The horizontal gray lines represent the scores of individual NMT systems used within the ensembles.

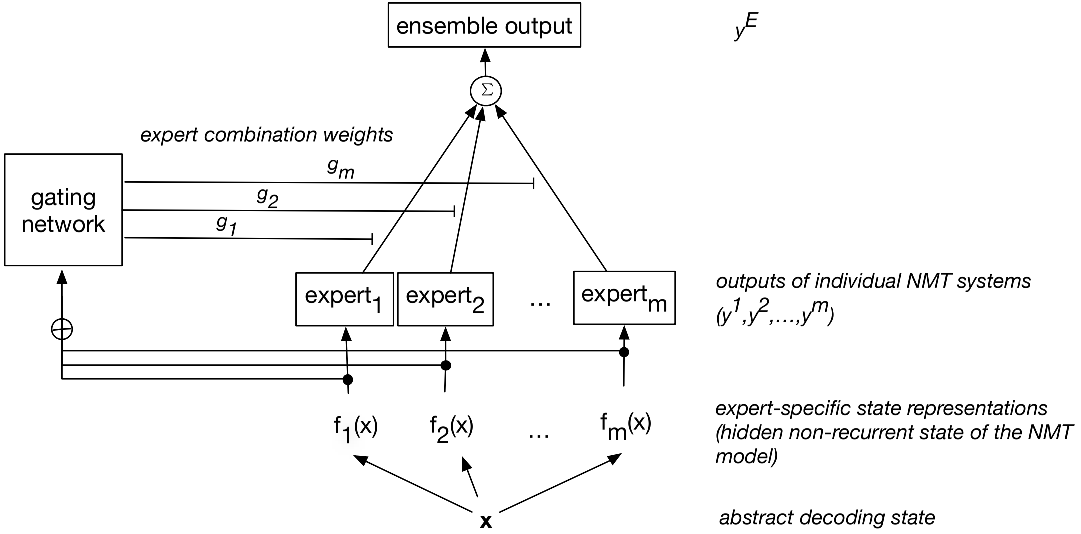


Figure 7.3: Mixture of NMT experts used to make context dependent translation prediction.

$$\mu = \sum_i g_i \mu_i, \tag{7.3}$$

where μ_i is the output of the i-th expert and g_i is its gating weight. Here, we realize context dependence by making use of a parameterized gating network.

Adapting the mixture of experts model (Jacobs et al., 1991) to the NMT scenario presents itself with a few challenges: NMT models are sequential, in the sense that the output at time step i depends on the current input word and the previous hidden state, which encodes the translation history for a *given* expert. This leads to two problems: the input representation is specific to an expert and it is also quite complex as it is a combination of hidden state and previously predicted word. We address the first problem by simply concatenating vectors which are inputs to each of the translators at time step i. There are a number of ways to address the second problem. Essentially, we would like to think of input x as some abstract decoding state corresponding to the context of the ensemble translation process at time step i.

In the first set of experiments, we opt for using the already available representations for the decoding state x, rather than formulating an explicitly, linguistically-motivated definition. Given the complex modular structure of an NMT model, there are a number of hidden states, such as the hidden recurrent states (*h*), the context vector (*c*), or the non-recurrent hidden state $\widetilde{h}$, which can be chosen to represent the decoder state which is the input to the gating network; see Section 2.2 for explanation of notation. In our approach, we use the last hidden state $\widetilde{h}$. We choose $\widetilde{h}$ because it already captures a large amount of information such as the previously predicted word, previous hidden state, and attention distribution over the source words.[3] In addition, the output layer is more directly connected to $\widetilde{h}$ than any of the states from lower layers. This is an important consideration given that the amount of training data is severely limited.

The architecture of our context-dependent combination function is presented graphically in Figure 7.3. The ensemble output $y^{\mathcal{E}}$ is computed as in Equation 7.4, where g_j is the gating weight, x represents the abstract decoding state at step i and $f^j(x)$ is its expert-specific representation (for expert j), namely $\widetilde{h}^j$:

$$y^{\mathcal{E}}(x_i) = \sum_j g_j \mu_j(x_i) \tag{7.4}$$

$$\mu_j(x_i) = \textit{softmax}(W_y f^j(x_i)) \tag{7.5}$$

$$f^j(x_i) = \widetilde{h}_i^j \tag{7.6}$$

$$g = \textit{softmax}(W_{\textit{gate}} h_{\textit{gate}} + b_{\textit{gate}}) \tag{7.7}$$

[3] In our preliminary experiments, we also experimented with using other layers of the NMT model as the decoding state x: the top-most recurrent layer of decoder, the context vector, and the embedding of the previously predicted target word as the decoding state. However, using the non-recurrent hidden state $\widetilde{h}$ achieved the best overall results.

Table 7.2: Translation results for individual NMT systems. For each language pair we trained four NMT systems with different weight parameter initializations. Decoding beam size is equal to 20. For each pair we provide the best score and the mean score with standard deviation.

system	BLEU	METEOR
de→ en best	20.58	49.16
de→ en mean	20.31 ± 0.34	48.88 ± 0.33
fr→ en best	27.80	56.91
fr→ en mean	27.03 ± 0.87	56.05 ± 0.82

$$h_{gate} = tanh(W_{hid}[f^1(x); ...; f^m(x)] + b_{hid}) \tag{7.8}$$

g_j is the j-th output unit of the gating network computed as in Equation 7.8. The gating network is a feed-forward neural network with one hidden layer of size 250 and a tanh non-linear activation function. The output layer is of size m, where m is the number of experts. Values of the output layer are normalized by applying *softmax*. The mixture model allows to back-propagate errors both to the gating network and the experts themselves. However, considering the small size of the training data and the complexity of the experts, in terms of number of parameters, full back-propagation is likely to result in over-fitting. Therefore, we only update the weights of the gating network, where the weights of the NMT predictors have been pre-trained separately.

7.4 Experiments

In Section 7.2 we have shown that NMT ensembles, and in particular multi-source ensembles, can improve translation quality. We proposed two methods to learn an ensemble combination function from data, which is more capable of exploiting the potential of ensembles than simple uniform weighting. In this section we validate these methods in translation experiments.

Translation results for individual systems comprising the ensembles are summarized in Table 7.2; see Section 6.3 for description of the data and NMT architecture. We compare the two ensemble induction methods, which are same-source systems with different parameter initializations and multi-source set, and apply all combination methods. For each ensemble set type, we evaluate ensembles of size 2 and size 4. All ensemble results are presented in Table 7.4.

As we explain in Section 6.3.2, we make sure, prior to training of the individual systems, that the output vocabularies of the two language pairs are the same. This will allow to do ensemble combination.

Notation. The notation below should be understood as follows: $de_k^{k+l} \rightarrow en$ stands for a single-source ensemble of l German-English systems. Analogously for French-English. *de,fr → en* is a multi-source ensemble of size 2, and we use subscript

Table 7.3: Comparison of individual German-English and French-English systems, empirical global combination upper bound (as estimated in Table 7.1), context-sensitive combination (Equation 7.4) and random mixture of experts combination systems. The latter result is an average over runs with different random seeds.

system	BLEU	METEOR
de → en	20.5	48.9
fr → en	27.8	56.9
fr, de → en global upper bound	30.2	58.9
fr, de → en context-sensitive combination	30.3	59.2
fr, de → en random gating combination	26.6	55.7

indices if there are more than 2 systems in an ensemble covering the same source language. We only apply context-dependent combination for ensembles of size 2 to avoid overfitting for bigger ensembles. Note that this does not prevent the application of our context-dependent combination method to bigger ensembles, as we can combine systems hierarchically.

In a hierarchical ensemble the set of predictors is divided into disjoint subsets and each of the subsets is combined separately. The resulting combination systems can then be treated as predictors in a new ensemble, and thus can be further combined for joint prediction. In our case the maximal number of predictors is 4, therefore our hierarchical ensembles have 2 levels. Hierarchical ensembles allow one to make prediction combinations multiple times which can further boost the potential of an ensemble. Since in this chapter we only consider a low-resource scenario with a small amount of training trilingual data, we do not train a hierarchical combination function. Instead, we perform global or context dependent combination for ensemble sets at the bottom level (level of individual NMT systems) and then weight their outputs uniformly (level of combined systems). We use curly brackets to denote hierarchical combination.

One can see in Table 7.4 that multi-source ensembles generally perform better that single-source ensembles. The largest improvements for multi-source ensembles are due to our context-dependent combination method. This result supports the hypothesis that the trained gating network is able to learn to differentiate between contexts where one of the individual systems performs better than the other. However, based on this result alone, it is not clear how much the observed improvement is due to the effectiveness of the gating network. Since we saw in our grid search experiment (Figure 7.2), even the most disadvantageous combination gives some improvements over individual predictors. So even if the gating network did not learn a lot, on average we are likely to get improvements. First of all, we note that a multi-source ensemble with contextual combination outperforms the empirical upper bound estimated in Section 7.2.2 of global combination, although the difference in BLEU is marginal. This suggests that the gating network at least is able to learn the average relative performance of the systems in the ensemble. Additionally, we ran an experiment for an ensemble of a German-English

Table 7.4: Translation experiments for the French, German into English scenario. *Ensemble set* designates ensemble induction method. *Combination type* refers to the method used to combine predictions during decoding. We use curly brackets to denote hierarchical ensemble combination.

	Combination type					
Ensemble set	uniform		global		context-dependent	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
$\mathrm{de}_1^2 \rightarrow$ en	21.8	50.4	21.8	50.4	21.8	50.3
$\mathrm{de}_1^4 \rightarrow$ en	21.8	50.4	21.8	50.4	-	-
$\{\mathrm{de}_1^2, \mathrm{de}_3^4\} \rightarrow$ en	-	-	21.8	50.4	**22.8**	**51.0**
$\mathrm{fr}_1^2 \rightarrow$ en	29.2	58.0	29.2	58.1	29.3	58.1
$\mathrm{fr}_1^4 \rightarrow$ en	29.2	58.0	29.2	58.1	-	-
$\{\mathrm{fr}_1^2, \mathrm{fr}_3^4\} \rightarrow$ en	-	-	29.2	58.1	**30.2**	**59.0**
de,fr→ en	**29.5**	**58.3**	**29.9**	**58.7**	30.3	59.2
de_1,de_2,fr_1,$\mathrm{fr}_2 \rightarrow$ en	29.4	58.3	29.3	58.2	-	-
$\{\{\mathrm{de}_1,\mathrm{fr}_1\},\{\mathrm{de}_2,\mathrm{fr}_2\}\}\rightarrow$ en	-	-	29.2	57.9	**31.5**	**60.3**

and a French-English systems whereby the gating weights were sampled uniformly (under the constraint that they are positive and sum up to 1). The results are summarized in Table 7.3. We can see that the randomly mixed ensemble obtains substantially lower scores than either the global upper bound or the context-sensitive combination. The results are also lower than the French-English system, but outperform the German-English system. It is likely that in a subset of prediction steps the random gating weights were "correct", thus explaining the fact that the randomly mixed system is stronger than the weaker system in the ensemble.

On the other hand, for single-source ensembles, context-dependent combination (by itself) does not provide additional improvements as compared to global weighting. This suggests that the variation found in single-source ensembles is not as systematic as in multi-source ensembles. As future work, one could perform a more linguistically oriented analysis to identify contexts triggering a high degree variation in ensembles. The results of such analysis will also provide the basis for a more linguistically oriented definition of the decoding state x as defined in Section 7.3.

Previous approaches using NMT ensembles often report performance increases for ensembles consisting of a larger number of systems, typically 8 or 12. One could therefore speculate that ensembles of 4 systems are not enough to significantly increase diversity as compared to an ensemble of size 2. Of course, our result are also influenced by several other factors such as the choice of languages, training data, etc. However, we should point out that hierarchically combining a set of 4 systems does improve translation quality. At this point, hierarchical combination still requires further investigation, but for the time being, it can be seen as a simple approach to further improve translation quality.

7.5 Conclusions

In this chapter we compared existing ensemble set induction methods for NMT and proposed two general system combination methods: global (across instances) weighting of predictors' outputs and context-dependent weighting. Our main goal was to answer **RQ3**: that translation systems from different source languages into the same target language have complementary strengths and weaknesses in terms of translation performance. We also introduce an approach that can exploit the respective strengths and weaknesses to achieve better translation quality. In our experiments with German-English and French-English we found that multi-source ensembles yield the best performance, compared to the individual translation systems, as well as compared to single-source ensembles of NMTs produced by different random initializations. This is an interesting finding because individually the two systems differ substantially in their translation quality. We also found that ensemble combination based on a gating network that decides how to combine systems at every prediction step achieves better performance as compared to a global (constant) combination function or uniform weighting in the majority of cases.

We can now revisit **RQ3**:

RQ3 Can we exploit the variation in cross-lingual correspondence and improve translation quality with multi-source NMT ensembles?

RQ3.a. How does ensemble performance depend on the quality of individual translation systems that are part of it?

Our experiments with multi-source ensembles showed that a substantially weaker NMT system (as evaluated by BLEU and METEOR) still improves translation quality when added to a strong NMT system. At the same time, monolingual ensembles of strong NMT systems with approximately the same quality actually produced lower quality translation output. This stresses the importance of diversification in NMT ensembles. The fact that there is no straightforward dependence between individual quality and ensemble quality in NMT is also supported by previous literature (Sennrich and Haddow, 2016), where ensemble combination is done over different training snapshots of the same system.

RQ3.b. Is there systematicity in what a neural translation system for a given language pair is good at, and what aspects of the target side it reproduces suboptimally?

We proposed to train a context-dependent combination function for NMT ensembles, which decides on the combination weights for each predicted target word separately. This method increased translation output quality for both monolingual and multi-source ensembles. Increasing metric scores for monolingual ensembles suggests that the combination function is able to learn even subtle distinctions in prediction behavior between models produced by different initializations. The best results overall were obtained by applying this combination method to multi-source ensembles. We stress that this ensemble method produced scores higher than the weaker system by almost 10 BLEU and 9 METEOR. This results implies that there is substantial difference in the prediction behaviors of NMT systems for different source languages and our combination learning method is capable of uncovering it.

RQ3.c. How do multi-source ensembles compare to ensembles of NMT systems for single language pair trained with different initialization seed?

The translation experiments in this chapter demonstrated that having different source languages yields an NMT ensemble diversification method which can be used to obtain improvements in translation quality. Our results suggest that it is a better source of diversity than monolingual ensembles. We also proposed and evaluated ensemble combination methods that can implicitly capture the diversity and learn to differentiate between local contexts where each individual system is likely to contribute more to on optimal prediction. Future work could include research on how our general knowledge about linguistic differences between some given languages relate to multi-source ensemble performance with these source languages.

Overall, our findings about multi-source ensembles is a compelling result and it leaves us with a number of questions for future work. First, can we characterize linguistically what types of contexts are mores suited to be translated by a German-English system, and which are more suited to be translated by a French-English system?

Gaining insights in that direction can help us answer another question: is there a better way to represent the current context which is the input to the gating network? In this chapter we used a concatenation of each system's last hidden state $\widetilde{h}$, but a potentially more effective and linguistically more intuitive representation may be found. Finally, it would be interesting to see to what extent our approach can benefit from three or more mutually different source languages.

8 Conclusions

This is the final chapter of the thesis. In Section 8.1 we review our research questions and the main findings of the thesis. In Section 8.2 we discuss the limitations of our methods when answering the research questions and discuss some directions for future work.

8.1 Main findings

The central topic of this thesis is the exploration of properties that are common across languages and properties that differentiate them, in the context of machine translation. The field of machine translation aims to render the content expressed in one language into another language. Therefore, the difficulty of this task is proportional to how different the given two languages are in expressing the same information. In the first part of the thesis, we focus specifically on the structures that each given language uses to express the same semantics. The core research goal of the first part of the thesis is to validate the hypothesis that every language has a level of representation which is shared, to some extent, by all languages. This level of representation is the syntactic structure of a sentence, i.e., the way pieces of information are grouped together to form an utterance. The implication of this hypothesis is that one can narrow down the search for translation correspondences between units of any two languages, as it must to some extent be consistent with the syntactic structure of both languages. In the second part of the thesis, we make use of the linguistic idea that despite the observation that all natural languages share certain ways of expressing content, there are also many features that can account for the observed diversity of languages. We adopt the hypothesis that the differences between languages are systematic, which implies that the task of learning the translation correspondence for two different language pairs with one common language will be difficult in different ways.

These are very broad ideas, which have been explored before in machine translation and natural language processing. In this thesis we narrowed down their exploration by proposing specific models that modify existing machine translation baselines by incorporating these ideas about the nature of translation correspondence. We investigated them in a series of research questions. Below we revisit the research questions, explaining how we addressed these questions and what findings we obtained.

RQ1 Can we improve reordering by modeling sequences of syntactic structures representing basic operational units of translation?

To address this question, we proposed to extend an existing model of translational correspondence, bilingual language models (BiLMs), which model sequences of elementary translational units. The concrete version of BiLMs that we worked with defined the translational units to be minimal, i.e., non-decomposable. We proposed a new representation for BiLM tokens, which incorporates parts of a source dependency tree corresponding to the words in a given bilingual token. Our idea was to devise a more abstract representation, but also to use complex syntactic annotation, which is likely to capture word ordering regularities and is also likely to be similar across languages (given the hypothesis discussed above) thus capturing reordering regularities. We integrated this model as a feature in the log-linear combination in a phrase-based SMT system.

RQ1.a. Can the representations only include the local syntactic information of a node in a syntactic parse? What is the minimum context that the local representation should incorporate?

DepBiLMs are BiLMs with local syntactic representations, defined in terms of the immediate vicinity of a node in a dependency tree. Our extensive experiments showed that depBiLMs improve translation quality overall, and reordering in particular, as demonstrated by reordering-sensitive metrics and translation experiments with an increased distortion limit. We compared the translation performance of depBiLMs to BiLMs with simple POS-based representations. The latter showed some of the worst results, only barely improving upon the baseline. From this we can conclude that there should be a certain degree of specificity in the syntactic representation to improve translation.

RQ1.b. How do local syntactic representations compare to representations including explicit lexical information of the basic translational units?

In general, depBiLMs produced translations at least as good as the one produced by lexicalized BiLMs. We have found some indications that they in fact capture complementary phenomena (at least for Arabic-English), whereby Lex•Lex is better at more specific and short-term reordering, while depBiLMs capture more general patterns of translation correspondence. We also note that even though depBiLMs were trained on a substantially smaller training corpus, they were able to achieve the same or better level of generalization (which is expected given the smaller and more abstract vocabulary).

RQ1.c. What kind of reordering phenomena are captured by such models?

As we mentioned before, depBiLMs tend to be good at reordering in general. We could find evidence of this model to perform especially well at some specific subclass of reordering patterns. But given the abstract definition of the depBiLM representations, it is expected that it would be more of a ‘universal’ reordering model, while more fine-grained lexicalized BiLMs capture better reordering phenomena for specific classes of words.

Overall, the findings relevant to **RQ1** entail that it may be sufficient to only specify local syntactic structure of words to capture regularities about the overall word order of a sentence. Moreover, this syntactic information is transferrable via translation correspondence, to capture reordering patterns. The idea of using local syntactic information to capture translational regularities has recently been reinterpreted in the context of NMT (Bastings et al., 2017).

The next research question is not about whether syntactic structures help express certain translational phenomena, but how similar syntactic structures for translational equivalents are.

RQ2 Is there a systematic mapping between source and target syntactic representations in a parallel sentence and can it be used to improve translation?

We address this very general question by assuming that there is such a mapping and by using sub-sentential translation correspondence (word alignments) to derive target language structures related to given source syntactic structures (obtained from a language-specific parser). We train a syntactic language model on the derived target-side parser and test whether this model, when incorporated as a feature in a phrase-based system, helps improve translation quality.

RQ2.a. Is there a universal characterization of a mapping between source and target structure? Can this characterization be used to constrain the decoding process to produce better translations?

In order to answer this question, we implemented and tested an existing model realizing such an approach, namely the syntactic cohesion model. Including it in the form of soft constraint features did not result in an improvement of translation quality for either of the language pairs. This entails that using ad-hoc assumptions about the nature of the source-target syntactic mapping may be too crude. Instead, we propose our own approach, where we first separately derive a mapping with elementary projection rules and then learn a statistical syntactic language model over it.

RQ2.b. Can the mapping be defined in terms of projection constraints between *elementary* parts of source and target structures? Can we fit a statistical model over the resulting corresponding source and target structures to characterize the overall mapping?

We proposed a simple set of rules, which projects a source structure onto the target. We used a structured language model to learn a characterization of target sequences parsed in this fashion (a model called bilingual structured language model, BiSLM). The model yielded statistically significant improvements for both Arabic-English and Chinese-English under rescoring and decoding settings. This result suggests that the BiSLM indeed captures the systematic relation between source and target syntactic structures. This correspondence can be used as a feature to improve translation quality, as measured by improvements in translation evaluation metrics. At the same time, we have some evidence that the models in their current form are too expressive and also accept incorrect translation hypotheses.

RQ2.c. What are the important mapping constraints that result in structured language models improving translation output?

We found that the parameter regulating how to incorporate unaligned target words had the most effect. The parameter regulating how to deal with non-cohesive translation did not appear to have an effect, suggesting that translations (for our given baselines) actually tend to be cohesive. This is also a potential explanation for why the cohesive constraints from **RQ2.a** did not contribute to translation quality. Finally, we experimented with a few labeling strategies and found that no or very simple labeling produced better metric scores.

We must note that the idea of deriving syntactic structure of a language based on syntactic correspondence is not entirely new and underlies (in a somewhat different form) a particular line of research in syntax-based SMT (Wu, 1997b; Chiang, 2007; Stanojevic, 2015). However, we showed that it is not necessary for syntax to be at the center of a translation framework, and that it can be useful as a feature of an essentially sequential phrase-based SMT. This finding is encouraging for other MT frameworks, such as NMT, and in fact recently a model has been proposed that incorporates a mechanism into a sequence-to-sequence NMT model which keeps track of how much the source tree has been attended to during decoding (Chen et al., 2017).

The next research question is answered in the second part of the thesis, which aims to exploit the systematic diversity among languages and language pairs.

RQ3 Can we exploit the variation in cross-lingual correspondence and improve translation quality with multi-source NMT ensembles?

We apply the assumption that systematic differences between language pairs can induce diversity into an ensemble set of NMT systems. Diversity is a necessary property for an ensemble set to improve the quality of individual predictions. We assume that relatively different languages (as follows from linguistic theory) are likely to produce systematically different kinds of translations when being used as source sides in translation systems with a common target side.

RQ3.a. How does ensemble performance depend on the quality of individual translation systems that are part of it?

Our experiments with multi-source ensembles showed that a substantially weaker NMT system (as evaluated by BLEU and METEOR) still improves translation quality when added to a strong NMT system. At the same time, monolingual ensembles of strong NMT systems with approximately the same quality actually produced lower quality translation outputs. This stresses the importance of diversification in NMT ensembles. The fact that there is no straightforward dependence between individual quality and ensemble quality in NMT is also supported by previous findings in the literature (Sennrich and Haddow, 2016), where ensemble combination is done over different training snapshots of the same system.

RQ3.b. Is there systematicity in what a neural translation system for a given language pair is good at, and what aspects of the target side it reproduces suboptimally?

We proposed to train a context-dependent combination function for NMT ensembles, which decides on the combination weights for each predicted target word separately. This method increased translation output quality for both monolingual and multi-source ensembles. Increasing the metric scores for monolingual ensembles suggests that the combination function is able to learn even subtle distinctions in prediction behavior between models produced by different initializations. The best results overall were obtained by applying this combination method to multi-source ensembles. We stress that this ensemble method produced scores higher than the weaker system by almost 10 BLEU and 9 METEOR. These results imply that there is a substantial difference in the prediction behaviors of NMT systems for different source languages and our combination learning method is capable of exploiting it.

RQ3.c. How do multi-source ensembles compare to ensembles of NMT systems for a single language pair trained with different initialization seeds?

The translation experiments in this chapter demonstrated that having different source languages yields an NMT ensemble diversification method which can be used to obtain improvements in translation quality. Our results suggest that it is a better source of diversity than monolingual ensembles. We also proposed and evaluated ensemble combination methods that can implicitly capture the diversity and learn to differentiate between local contexts where each individual system is likely to contribute more to an optimal prediction. Future work could include research on how our general knowledge about linguistic differences between some given languages relate to multi-source ensemble performance with these source languages.

The results of our multi-source ensemble experiments are encouraging and suggest a simple way to improve translation quality, under the assumption that multi-parallel source text is available at decoding. Moreover, this method of ensemble diversification can be applied to other NLP tasks as well which involve extracting information from an input text (again under the same assumption). Such tasks may include: summarization, question answering, and sentiment analysis.

8.2 Future work

In the previous section we revisited the research questions addressed in this thesis and summarized the findings from the experiments designed to answer those questions. Also, we outlined some general research directions which our findings support.

However, even though we did manage to obtain useful insights regarding our questions, the design of our experiments is by no means exhaustive in the sense that they give complete answers to all research questions. In this section we list limitations of our work and discuss ways to address these limitations as part of future work. We group the limitations by what aspect of the experimental setting needs to be addressed.

Datasets. As we pointed out in Chapter 4, the different BiLM models (with different representations) were trained on training data sets of different sizes, due to the source parser not producing a well-formed annotation for all of the sentences in the training set. Even though our proposed model (depBiLM) showed comparable or better performance in the experiments, it is also useful to compare the model when they have access to the same amounts of data.

Note that our experimental setting in Chapter 7 did not suffer from this shortcoming as we made sure that the training sizes for all individual NMT systems are approximately the same. However, we worked with a relatively small training data set (see Section 6.3 (Cettolo et al., 2012)). It would also be important to see whether the benefits of our multi-source ensembling method are helpful when individual systems use substantially larger data sets to train on. We performed a series of experiments (not reported in Chapter 7) with multi-source systems (German, French, Spanish into English) trained on WMT data combined via uniform averaging, which resulted in significant improvements for all language pair combinations. This indicates that the diversity among systems for different language pairs is preserved also in a large data scenario and encourages experiments with more advanced combination methods. Another limitation of Chapter 7 is that despite the fact that the chosen language pairs were relatively diverse, it would be interesting to see the effect of ensembles for even more distant languages, such as Chinese and French as source languages, for example.

Baselines and comparison systems. Having a stronger baseline will always provide stronger support for the effectiveness of a proposed model and its usefulness in real word scenarios. For example, in Chapters 4 and 5 one could include more reordering models into the baseline, such as (Galley and Manning, 2008). As for Chapter 7, we have already mentioned above that having systems trained on a larger data set would help to further demonstrate the effectiveness of the method. Having a more diverse set of comparison systems, i.e., systems implementing alternative methods, would also contribute to a stronger experimental setup. In Chapter 4 it would be interesting to study the performance of BiLMs with rich linguistic annotations of a different kind, such as in (Crego and Mariño, 2006), although the application of such models is limited by the availability of the annotation resources. An interesting comparison to the BiSLM from Chapter 5 would be to train a monolingual target-language structured model. We have done some preliminary experiments in this direction in a rescoring scenario, and the result was that our BiSLM achieved better results. We think it has to do with the fact that the translation hypotheses to be parsed were not well-formed target sentences, as a result of which the parser did not produce reasonable structural annotations. As for Chapter 7 on multi-source ensembles, we could compare our model to additional alternative diversification, such as Bagging or different stages of a model's training, and combination, such as computing a belief function for each class that can be used for comparison; see the methods discussed in Section 6.1.

Additional experiments under the original setting. Even if we did not address the limitations above, still additional experiments could have been run in the original experimental setup. In Chapter 5 we use the rescoring experiments as a way to select the models for the decoding experiments. However, in rescoring, the word alignments (with respect to which the source parse is projected) are determined by the baseline model, while during decoding the alignments can be influenced by the BiSLM itself. Therefore a more exhaustive set of decoding experiments would be beneficial for understanding the effect of the proposed model. Also, in Chapter 7 we limit the size of ensemble sets to four; while it would have been interesting to investigate how the effect of the discussed ensemble methods changes with the growing size of the ensemble set.

Assumptions about what is given at input. The first two research chapters propose models relying on precomputed parses of the source sentence. Good parsers are available for some popular machine translation languages (English, French, German, Chinese, Arabic), however, for many languages, parsers are of poor quality or not even available at all. One way to address this limitation is to see investigate our proposed models perform when the source parser is of low quality, for example, when trained on projected annotations from another language (McDonald et al., 2011). A more extreme approach to address this limitation would be to test whether we can still use the models where the structural annotations are induced in an unsupervised manner, see (Klein and Manning, 2004). The major assumption necessary to apply our multi-source ensemble method is the availability of a multi-source text at test time and a small multi-lingual data set (including the target language) to train a combination function. We see two directions to weaken this assumption. One direction is to train a combination function for each language pair disjointly, by having a model of how confident a given system is at predicting in a given context. Another direction is to adapt the method to comparable corpora, which are available for more languages.

To summarize, in this thesis we have seen how common structural representations among languages and systematic differences between languages and language pairs can be used to enhance machine translation quality. With the suggestions that we have just given, we believe there is further potential for exploiting the universal linguistic properties and the systematic linguistic diversity to benefit machine translation.

Bibliography

A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986. (Cited on page 31.)

Y. Al-Onaizan and K. Papineni. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536, Sydney, Australia, 2006. Association for Computational Linguistics. (Cited on page 39.)

N. Bach, S. Vogel, and C. Cherry. Cohesive constraints in a beam search phrase-based decoder. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1–4. Association for Computational Linguistics, 2009. (Cited on pages 28, 71, 72, 81, and 83.)

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015. (Cited on pages 18, 19, 93, and 103.)

B. Bangalore, G. Bordel, and G. Riccardi. Computing consensus translation from multiple machine translation systems. In *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*, pages 351–354. IEEE, 2001. (Cited on page 99.)

J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017. (Cited on pages 19, 29, 65, 93, and 121.)

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003. (Cited on page 15.)

L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267. Association for Computational Linguistics, 2016. (Cited on page 17.)

A. Birch and M. Osborne. Lrscore for evaluating lexical and reordering quality in mt. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 327–332. Association for Computational Linguistics, 2010. (Cited on pages 22 and 48.)

A. Bisazza and M. Federico. A survey of word reordering in statistical machine translation: Computational models and language phenomena. *Computational Linguistics*, 42(2):163–205, 2016. (Cited on page 15.)

A. Bisazza and C. Monz. Class-based language modeling for translating into morphologically rich languages. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1918–1927, 2014. (Cited on page 47.)

O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics Baltimore, MD, USA, 2014. (Cited on pages 99, 104, and 105.)

O. Bojar, Y. Graham, A. Kamran, and M. Stanojevic. Results of the wmt16 metrics shared task. In *First Conference on Machine Translation (WMT16)*, pages 199–231, 2016. (Cited on page 20.)

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. (Cited on page 98.)

J. Bresnan, S. Dingare, and C. D. Manning. Soft constraints mirror hard constraints: Voice and person in english and lummi. In *Proceedings of the LFG01 Conference*, pages 13–32. Stanford: CSLI Publications. E. Verhoeven, 2001. (Cited on page 25.)

G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005. (Cited on pages 97 and 98.)

P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993. (Cited on pages 1, 3, 11, and 12.)

M. Carl, A. Way, and W. Daelemans. Recent advances in example-based machine translation. *Computational Linguistics*, 30(4):516–520, 2004. (Cited on page 1.)

M. Carpuat, Y. Marton, and N. Habash. Improving Arabic-to-English statistical machine translation by reordering post-verbal subjects for alignment. In *Proceedings of the Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 178–183. Association for Computational Linguistics, 2010. (Cited on pages 38 and 39.)

M. Cettolo, C. Girardi, and M. Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16^{th} Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May 2012. (Cited on pages 99, 100, and 124.)

P.-C. Chang and K. Toutanova. A discriminative syntactic word order model for machine translation. In

Proceedings of the Annual Meeting for the Association for Computational Linguistics, volume 45, page 9. Association for Computational Linguistics, 2007. (Cited on page 28.)

P.-C. Chang, H. Tseng, D. Jurafsky, and C. D. Manning. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59. Association for Computational Linguistics, 2009. (Cited on pages 34, 46, and 80.)

E. Charniak. Immediate-head parsing for language models. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2001. (Cited on page 30.)

C. Chelba and F. Jelinek. Structured language modeling. *Computer Speech and Language*, 14(4):283–332, 2000. (Cited on pages 6, 25, 30, 31, 32, 68, and 69.)

H. Chen, S. Huang, D. Chiang, and J. Chen. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2017. (Cited on pages 19, 93, and 122.)

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996. (Cited on pages 15 and 35.)

C. Cherry. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of Association for Computational Linguistics*, pages 72–80. Association for Computational Linguistics, 2008. (Cited on pages 28, 67, 71, 72, 78, and 81.)

D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007. (Cited on pages 28, 37, 103, and 122.)

D. Chiang. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452. Association for Computational Linguistics, 2010. (Cited on page 37.)

N. Chomsky. *Syntactic structures*. Walter de Gruyter, 2002. (Cited on page 25.)

J. H. Clark, C. Dyer, and A. Lavie. Locally non-linear learning for statistical machine translation via discretization and structured regularization. *Transactions of the Association for Computational Linguistics*, 2:393–404, 2014. (Cited on page 22.)

M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999. (Cited on pages 34, 46, and 80.)

J. M. Crego and N. Habash. Using shallow syntax information to improve word alignment and reordering for smt. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 53–61. Association for Computational Linguistics, 2008. (Cited on page 30.)

J. M. Crego and J. B. Mariño. Improving statistical mt by coupling reordering and decoding. *Machine Translation*, 20(3):199–215, 2006. (Cited on pages 30 and 124.)

J. M. Crego and F. Yvon. Factored bilingual n-gram language models for statistical machine translation. *Machine Translation*, 24(2):159–175, 2010a. (Cited on page 37.)

J. M. Crego and F. Yvon. Improving reordering with linguistically informed bilingual n-grams. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 197–205. Association for Computational Linguistics, 2010b. (Cited on pages 37 and 39.)

J. M. Crego et al. Reordered search, and tuple unfolding for ngram-based smt. In *In Proceedings of the MT Summit X*, 2005. (Cited on page 29.)

G. M. de Buy Wenniger and K. Sima'an. Hierarchical alignment decomposition labels for hiero grammar rules. In *SSST@NAACL-HLT*. Association for Computational Linguistics, 2013. (Cited on page 28.)

M. Denkowski and A. Lavie. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2011. (Cited on page 21.)

J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. M. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 1370–1380. Association for Computational Linguistics, 2014. (Cited on page 75.)

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 1:22, 1999. (Cited on page 103.)

Y. Ding and M. S. Palmer. Automatic learning of parallel dependency treelet pairs. In *In: Su KY., Tsujii J., Lee JH., Kwong O.Y. (eds) Natural Language Processing IJCNLP 2004. IJCNLP 2004. Lecture Notes in Computer Science*, volume 3248. Springer, 2004. (Cited on page 3.)

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. (Cited on page 109.)

N. Durrani, H. Schmid, and A. Fraser. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1045–1054. Association for Computational Linguistics, 2011. (Cited on pages 29 and 40.)

N. Durrani, A. Fraser, and H. Schmid. Model with minimal translation units, but decode with phrases. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–11. Association for Computational Linguistics, 2013. (Cited on page 30.)

C. Dyer. Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 406–414. Association for Computational Linguistics, 2009. (Cited on page 99.)

J. Eisner. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 205–208. Association for Computational Linguistics, 2003. (Cited on pages 3 and 28.)

D. Elliott and Á. Kádár. Imagination improves multimodal translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017. (Cited on page 17.)

J. Elming and N. Habash. Syntactic reordering for english-arabic phrase-based machine translation. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 69–77. Association for Computational Linguistics, 2009. (Cited on page 38.)

A. Eriguchi, K. Hashimoto, and Y. Tsuruoka. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2016. (Cited on pages 19, 29, and 65.)

O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho. Zero-resource translation with multi-lingual neural machine translation. *to appear in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016. (Cited on pages 104, 105, and 107.)

G. Foster, R. Kuhn, and H. Johnson. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61. Association for Computational Linguistics, 2006. (Cited on page 14.)

H. J. Fox. Phrasal cohesion and statistical machine translation. In *Proceedings the Conference on Empirical Methods in Natural Language Processing*, pages 304–311. Association for Computational Linguistics, 2002. (Cited on page 71.)

Y. Freund, R. E. Schapire, et al. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996)*, pages 148–156. Morgan Kaufmann, 1996. (Cited on page 98.)

M. Galley and C. D. Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856. Association for Computational Linguistics, 2008. (Cited on page 124.)

E. Garmash and C. Monz. Dependency-based bilingual language models for reordering in statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1689–1700. Association for Computational Linguistics, 2014. (Cited on pages 8 and 15.)

E. Garmash and C. Monz. Bilingual structured language models for statistical machine translation. In *Proceedings the Conference on Empirical Methods in Natural Language Processing*, pages 2398–2408. Association for Computational Linguistics, 2015. (Cited on page 8.)

E. Garmash and C. Monz. Ensemble learning for multi-source neural machine translation. In *Proceedings of the International Conference on Computational Linguistics*, 2016. (Cited on page 9.)

N. Ge. A direct syntax-driven reordering model for phrase-based machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 849–857. Association for Computational Linguistics, 2010. (Cited on pages 3, 28, and 67.)

D. Gildea. Dependencies vs. constituents for tree-based alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 214–221. Association for Computational Linguistics, 2004. (Cited on page 3.)

Y. Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017. (Cited on page 11.)

J. T. Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001. (Cited on page 15.)

S. Green and C. D. Manning. Better arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402. Association for

Computational Linguistics, 2010. (Cited on pages 34, 46, and 80.)

J. Gu, K. Cho, and V. O. Li. Trainable greedy decoding for neural machine translation. In *1st Workshop on Neural Machine Translation (and Generation)*. Association for Computational Linguistics, 2017. (Cited on page 19.)

J. Gubbins and A. Vlachos. Dependency language models for sentence completion. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2013. (Cited on pages 30 and 32.)

C. Gulcehre, O. Firat, K. Xu, K. Cho, and Y. Bengio. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148, 2017. (Cited on page 93.)

J. Havelka. Beyond projectivity: Multilingual evaluation of constraints and measures on non-projective structures. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 608–615. Association for Computational Linguistics, 2007. (Cited on page 69.)

M. Hopkins and J. May. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362. Association for Computational Linguistics, 2011. (Cited on pages 17 and 35.)

L. Huang and H. Mi. Efficient incremental decoding for tree-to-string translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 273–283. Association for Computational Linguistics, 2010. (Cited on pages 37 and 38.)

L. Huang, K. Knight, and A. Joshi. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–226, 2006. (Cited on pages 2, 3, 28, and 37.)

P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer. Attention-based multimodal neural machine translation. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, 2016. (Cited on page 17.)

R. Hwa, P. Resnik, A. Weinberg, and O. Kolak. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 392–399. Association for Computational Linguistics, 2002. (Cited on pages 6, 69, and 70.)

R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(3):311–325, 2005. (Cited on page 70.)

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. (Cited on pages 98, 109, and 112.)

N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2013. (Cited on pages 1, 17, and 103.)

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations*, 2015. (Cited on page 19.)

D. Klein and C. D. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 478. Association for Computational Linguistics, 2004. (Cited on page 125.)

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE, 1995. (Cited on page 15.)

K. Knight. Decoding complexity in word-replacement translation models. *Computational linguistics*, 25(4): 607–615, 1999. (Cited on page 14.)

P. Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 388–395, July 2004. (Cited on page 22.)

P. Koehn. *Statistical machine translation*. Cambridge University Press, 2009. (Cited on pages 1, 11, 13, and 16.)

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54. Association for Computational Linguistics, 2003. (Cited on pages 2, 13, 15, and 103.)

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007. (Cited on pages 34 and 35.)

M. Kuhlmann and J. Nivre. Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 507–514. Association for Computational Linguistics, 2006.

(Cited on page 69.)

L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003. (Cited on pages 97, 103, and 104.)

A. Lavie and M. J. Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115, 2009. (Cited on pages 21, 48, 81, and 102.)

A. Lavie, A. Parlikar, and V. Ambati. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation*, pages 87–95. Association for Computational Linguistics, 2008. (Cited on page 28.)

U. Lerner and S. Petrov. Source-side classifier preordering for machine translation. In *Proceedings of the Empirical Methods in Natural Language Processing*, 2013. (Cited on pages 3 and 28.)

J. Li, Z. Tu, G. Zhou, and J. van Genabith. Head-driven hierarchical phrase-based translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 33–37. Association for Computational Linguistics, 2012. (Cited on page 28.)

W. Ling, I. Trancoso, C. Dyer, and A. W. Black. Character-based neural machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2015. (Cited on pages 18, 19, and 103.)

Y. Liu, Q. Liu, and S. Lin. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 609–616. Association for Computational Linguistics, 2006. (Cited on pages 28 and 37.)

Y. Liu, Y. Huang, Q. Liu, and S. Lin. Forest-to-string statistical translation rules. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, page 704. Association for Computational Linguistics, 2007. (Cited on page 2.)

T. Lucien. Eléments de syntaxe structurale. *Paris, Klincksieck*, 1959. (Cited on page 25.)

M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics, 2015a. (Cited on page 100.)

T. Luong, M. Kayser, and C. D. Manning. Deep neural language models for machine translation. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, 2015b. (Cited on pages 15 and 19.)

J. B. Marino, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-Jussà. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549, 2006. (Cited on pages 29 and 37.)

Y. Marton and P. Resnik. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of the Association for Computational Linguistics*, pages 1003–1011, 2008. (Cited on pages 28 and 37.)

E. Matusov, N. Ueffing, and H. Ney. Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In *European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006. (Cited on pages 5, 99, and 103.)

R. McDonald, S. Petrov, and K. Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics, 2011. (Cited on page 125.)

W. Monroe, S. Green, and C. D. Manning. Word segmentation of informal arabic with domain adaptation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 2014. (Cited on page 33.)

C. Monz. Statistical machine translation with local language models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 869–879. Association for Computational Linguistics, 2011. (Cited on page 80.)

R. Moot and C. Retoré. *The Logic of Categorial Grammars - A Deductive Account of Natural Language Syntax and Semantics*, volume 6850 of *Lecture Notes in Computer Science*. Springer, 2012. (Cited on page 25.)

D. S. Munteanu and D. Marcu. Processing comparable corpora with bilingual suffix trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 289–295. Association for Computational Linguistics, 2002. (Cited on page 1.)

T. Naseem, R. Barzilay, and A. Globerson. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 629–637. Association for Computational Linguistics, 2012. (Cited on page 70.)

J. Niehues, T. Herrmann, S. Vogel, and A. Waibel. Wider context by using bilingual language models in machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206. Association for Computational Linguistics, 2011. (Cited on pages 6, 25, 29, 30, 37, 38, 39, 40, 41, 44, 47, and 58.)

S. Nirenburg. Knowledge-based machine translation. *Machine Translation*, 4(1):5–24, 1989. (Cited on page 1.)

E. W. Noreen. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley-Interscience, 1989. (Cited on page 22.)

F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003. (Cited on page 17.)

F. J. Och and H. Ney. Statistical multi-source translation. In *Proceedings of Machine Translation Summit*, volume 8, pages 253–258. Association for Computational Linguistics, 2001. (Cited on pages 5, 98, and 103.)

F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 295–302. Association for Computational Linguistics, 2002. (Cited on page 12.)

F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003a. (Cited on pages 13, 22, and 34.)

F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003b. (Cited on page 13.)

T. Osborne. Major constituents and two dependency grammar constraints on sharing in coordination. *Linguistics*, 46(6):1109–1165, 2008. (Cited on page 69.)

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics, 2002. (Cited on pages 20, 48, 81, and 102.)

A. Pauls and D. Klein. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 959–968. Association for Computational Linguistics, 2012. (Cited on page 30.)

M. Post and D. Gildea. Parsers as language models for statistical machine translation. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 172–181. Association for Computational Linguistics, 2008. (Cited on page 67.)

C. Quirk and A. Menezes. Dependency treelet translation: The convergence of statistical and example-based machine translation? *Machine Translation*, 20:43—65, March 2006. (Cited on pages 2, 70, 72, 74, and 75.)

M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2015. (Cited on page 20.)

S. Riezler and J. 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*. Association for Computational Linguistics, 2005. (Cited on page 22.)

B. Roark. Probabilistic top-down parsing and language modeling. *Computational linguistics*, 27(2):249–276, 2001. (Cited on page 30.)

L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010. (Cited on pages 98, 103, and 106.)

A.-V. Rosti, N. F. Ayan, B. Xiang, S. Matsoukas, R. Schwartz, and B. Dorr. Combining outputs from multiple machine translation systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235, 2007. (Cited on page 99.)

J. Schroeder, T. Cohn, and P. Koehn. Word lattices for multi-source translation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 719–727. Association for Computational Linguistics, 2009. (Cited on pages 5, 99, and 103.)

L. Schwartz. Multi-source translation methods. In *Proceedings of the Association for Machine Translation in the Americas 2008*. Association for Computational Linguistics, October 2008. (Cited on pages 5 and 103.)

L. Schwartz, C. Callison-Burch, W. Schuler, and S. Wu. Incremental syntactic language models for phrase-based translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 620–631. Association for Computational Linguistics, 2011. (Cited on page 67.)

R. Sennrich and B. Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91. Association for Computational Linguistics,

2016. (Cited on pages 104, 105, 117, and 122.)

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2015. (Cited on page 18.)

L. Shen, J. Xu, and R. M. Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the Association for Computational Linguistics*, pages 577–585. Association for Computational Linguistics, 2008. (Cited on pages 2, 28, 37, and 44.)

X. Shi, I. Padhi, and K. Knight. Does string-based neural mt learn source syntax? In *Proceedings of the Empirical Methods for Natural Language Processing*. Association for Computational Linguistics, 2016. (Cited on page 65.)

S. M. Shieber. Probabilistic synchronous tree-adjoining grammars for machine translation: The argument from bilingual dictionaries. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 88–95. Association for Computational Linguistics, 2007. (Cited on page 2.)

S. Shilen. Multiple binary tree classifiers. *Pattern Recognition*, 23(7):757–763, 1990. (Cited on page 98.)

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–231. Association for Computational Linguistics, 2006. (Cited on pages 21, 48, and 81.)

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014. (Cited on page 100.)

M. Stanojevic. Reordering grammar induction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 44–54. Association for Computational Linguistics, 2015. (Cited on pages 28 and 122.)

A. Stolcke, J. Zheng, W. Wang, and V. Abrash. Srilm at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, page 5, 2011. (Cited on pages 35, 46, and 80.)

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014. (Cited on pages 1, 5, 17, 18, 100, 103, and 107.)

C. Tillmann. A unigram orientation model for statistical machine translation. In *Proceedings of of the North American Chapter of the Association for Computational Linguistics*, pages 101–104. Association for Computational Linguistics, 2004. (Cited on page 15.)

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180. Association for Computational Linguistics, 2003. (Cited on pages 34, 46, and 80.)

H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*, volume 171, 2005. (Cited on page 33.)

Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2016. (Cited on pages 19 and 93.)

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. (Cited on page 19.)

C. Wang, M. Collins, and P. Koehn. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 737–745. Association for Computational Linguistics, 2007. (Cited on page 38.)

T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2007. (Cited on page 17.)

S. Wiseman and A. M. Rush. Sequence-to-sequence learning as beam-search optimization. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016. (Cited on pages 19 and 20.)

I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *Ieee transactions on information theory*, 37(4):1085–1094, 1991. (Cited on page 15.)

D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. (Cited on pages 98 and 103.)

D. Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403, 1997a. (Cited on page 3.)

D. Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403, 1997b. (Cited on pages 2, 28, 70, and 122.)

B. Xiang, N. Ge, and A. Ittycheriah. Improving reordering for statistical machine translation with smoothed priors and syntactic features. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 61–69. Association for Computational Linguistics, 2011. (Cited on page 3.)

J. Xu, E. Matusov, R. Zens, and H. Ney. Integrated chinese word segmentation in statistical machine translation. In *International Workshop on Spoken Language Translation (IWSLT) 2005*, 2005. (Cited on page 99.)

K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics, 2001. (Cited on pages 2, 3, 28, 68, 70, and 72.)

H. Yu, H. Mi, L. Huang, and Q. Liu. A structured language model for incremental tree-to-string translation. *Proceedings of the 25th International Conference on Computational Linguistics*, pages 1133–1143, 2014. (Cited on pages 68 and 72.)

A. Zollmann and S. Vogel. A word-class approach to labeling pscfg rules for machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1–11. Association for Computational Linguistics, 2011. (Cited on page 28.)

Summary

The central topic of this thesis is the exploration of properties that are common across languages and properties that differentiate them, in the context of machine translation. The field of machine translation aims to render the content expressed in one language into another language. Therefore, the difficulty of this task is proportional to how different the given two languages are in expressing the same information.

In the first part of the thesis, we focus on properties common across languages. The core research goal of the first part of the thesis is to validate the hypothesis that every language has a level of representation which is shared, to some extent, by all languages. This level of representation is the syntactic structure of a sentence, i.e., the way pieces of information are grouped together to form an utterance. The implication of this hypothesis is that one can narrow down the search for translation correspondences between units of any two languages, as it must to some extent be consistent with the syntactic structure of both languages. We realize these ideas in the form of bilingual syntactic language models which are used as soft constraints during the translation process. Specifically, our first proposed model extends bilingual language models by adding a new kind of representations of bilingual tokens, based on dependency tree annotations of the source sentence. The experiments demonstrate that the proposed model improves translation quality, as well as improvements in reordering. Our second proposed model is a structured language model adapted to a bilingual scenario. We use the idea that there exists some systematic correspondence between source and target sentential structures and project source dependency annotations onto the target sentences to obtain representations for a structured language model. The model provides improvements in machine translation quality in a series of rescoring and decoding experiments.

In the second part of the thesis, we make use of the linguistic idea that despite the observation that all natural languages share certain ways of expressing content, there are also many features that differentiate them. We adopt the hypothesis that the differences between languages are systematic, which implies that the task of learning the translation correspondence for two different language pairs with one common language will be difficult in different ways. We make use of this idea by designing ensembles of neural machine translation systems sharing the target language but differing in their source languages (multi-source ensembles). The difference in source languages introduces the level of ensemble diversity necessary to improve an individual system's translation performance. We perform a comparison to monolingual ensembles obtained by initializing systems with different random seeds and observe systematically better performance for the multi-source ensembles.

Samenvatting

Het centrale onderwerp van dit proefschrift is het verkennen van gemeenschappelijke en onderscheidende eigenschappen van talen, in het kader van automatisch vertalen. Het veld van automatisch vertalen heeft als doel om de inhoud uitgedrukt in één taal in een andere taal om te zetten. De moeilijkheid van deze taak is evenredig met de mate van verschil tussen de twee talen.

In het eerste deel van het proefschrift richten we ons op eigenschappen die in alle talen voorkomen. De voornaamste onderzoeksvraag van het eerste deel van het proefschrift is het valideren van de hypothese dat alle talen een gedeelde onderliggende representatie hebben. Dit niveau van representatie is de syntactische structuur van een zin, die de manier bepaald waarop zinsdelen worden samengevoegd om een zin te vormen. Het gevolg van deze hypothese is dat men het zoeken naar vertaalcorrecties tussen eenheden van twee talen kan verkleinen, aangezien het in zekere mate in overeenstemming moet zijn met de syntactische structuur van beide talen. We implementeren deze ideeën in de vorm van tweetalige syntactische taalmodellen die gebruikt worden als zachte restricties tijdens het vertaalproces. Ons eerste voorgestelde model breidt tweetalige taalmodellen uit door een nieuwe soort representatie van tweetalige symbolen toe te voegen, gebaseerd op annotaties van de syntactische boom van de bronzin. De experimenten tonen aan dat het voorgestelde model de kwaliteit van de vertaling verbetert. Ons tweede voorgestelde model is een gestructureerd taalmodel aangepast aan een tweetalig scenario. We baseren ons op het idee dat er een aantal systematische overeenkomsten bestaan in de zinsstructuur tussen de bron- en de doeltaal en we projecteren syntactische annotaties van de bronzinnen op de doelzinnen om representaties te verkrijgen voor een gestructureerd taalmodel. Het model levert verbeteringen op in de kwaliteit van de automatische vertaling in een reeks rescoring- en decoderings-experimenten.

In het tweede deel van het proefschrift steunen we op het idee uit de linguïstiek dat, ondanks dat alle natuurlijke talen gedeelde eigenschappen hebben, er ook veel eigenschappen zijn die de verschillende talen onderscheiden. We nemen de hypothese aan dat de verschillen tussen talen systematisch zijn. Dit heeft als gevolg dat het leren van een vertaalmodel voor twee verschillende taalparen met één gemeenschappelijke taal, op verschillende manieren moeilijk zal zijn. We ontwerpen multi-source ensembles van neurale automatische vertaalsystemen die de doeltaal delen, maar verschillen in hun brontaal. Het verschil in brontalen zorgt ervoor dat de modellen verschillende fouten maken en bijgevolg geeft de combinatie van de individuele systemen een betere vertaling dan de individuele systemen. Een vergelijking met ééntalige ensembles die verkregen werden door de modellen te initialiseren met andere willekeurige waarden toont aan dat de multi-source ensembles systematisch beter presteren.