Reordering in statistical machine translation

Yahyaei, Mohammad Sirvan

Reordering in Statistical Machine Translation

Mohammad Sirvan Yahyaei

University of London

Thesis submitted for the degree of Doctor of Philosophy
at Queen Mary, University of London

October 2011

Declaration of Originality

I hereby declare that this thesis and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

The material contained in this thesis has not been submitted, either in whole or in part, for a degree or a diploma or other qualifications at the University of London or any other university.

Some parts of this work have been previously published as:

- Sirvan Yahyaei and Christof Monz. Decoding by dynamic chunking for statistical machine translation. In *MT Summit XII: Proceedings of the Twelfth Machine Translation Summit*, pages 160–167, Ontario, Canada, August 2009
- Sirvan Yahyaei and Christof Monz. Dynamic distortion in a discriminative reordering model for statistical machine translation. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation*, IWSLT '10, pages 353–360, 2010a
- Sirvan Yahyaei and Christof Monz. The QMUL system description for IWSLT 2010. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation*, IWSLT '10, pages 157–162, 2010b
- Sirvan Yahyaei, Marco Bonzanini, and Thomas Roelleke. Cross-lingual text fragment alignment using divergence from randomness. In *Proceedings of the 18th edition of the International Symposium on String Processing and Information Retrieval (SPIRE)*, 2011

Sirvan Yahyaei

London, October 2011

Abstract

Machine translation is a challenging task that its difficulties arise from several characteristics of natural language. The main focus of this work is on *reordering* as one of the major problems in MT and statistical MT, which is the method investigated in this research. The reordering problem in SMT originates from the fact that not all the words in a sentence can be consecutively translated. This means words must be skipped and be translated out of their order in the source sentence to produce a fluent and grammatically correct sentence in the target language. The main reason that reordering is needed is the fundamental word order differences between languages. Therefore, reordering becomes a more dominant issue, the more source and target languages are structurally different.

The aim of this thesis is to study the reordering phenomenon by proposing new methods of dealing with reordering in SMT decoders and evaluating the effectiveness of the methods and the importance of reordering in the context of natural language processing tasks. In other words, we propose novel ways of performing the decoding to improve the reordering capabilities of the SMT decoder and in addition we explore the effect of improving the reordering on the quality of specific NLP tasks, namely named entity recognition and cross-lingual text association. Meanwhile, we go beyond reordering in text association and present a method to perform cross-lingual text fragment alignment, based on models of divergence from randomness.

The main contribution of this thesis is a novel method named dynamic distortion, which is designed to improve the ability of the phrase-based decoder in performing reordering by adjusting the distortion parameter based on the translation context. The model employs a discriminative reordering model, which is combining several fea-

tures including lexical and syntactic, to predict the necessary distortion limit for each sentence and each hypothesis expansion. The discriminative reordering model is also integrated into the decoder as an extra feature. The method achieves substantial improvements over the baseline without increase in the decoding time by avoiding reordering in unnecessary positions.

Another novel method is also presented to extend the phrase-based decoder to dynamically chunk, reorder, and apply phrase translations in tandem. Words inside the chunks are moved together to enable the decoder to make long-distance reorderings to capture the word order differences between languages with different sentence structures.

Another aspect of this work is the task-based evaluation of the reordering methods and other translation algorithms used in the phrase-based SMT systems. With more successful SMT systems, performing multi-lingual and cross-lingual tasks through translating becomes more feasible. We have devised a method to evaluate the performance of state-of-the art named entity recognisers on the text translated by a SMT decoder. Specifically, we investigated the effect of word reordering and incorporating reordering models in improving the quality of named entity extraction.

In addition to empirically investigating the effect of translation in the context of cross-lingual document association, we have described a text fragment alignment algorithm to find sections of the two documents in different languages, that are content-wise related. The algorithm uses similarity measures based on divergence from randomness and word-based translation models to perform text fragment alignment on a collection of documents in two different languages.

All the methods proposed in this thesis are extensively empirically examined. We have tested all the algorithms on common translation collections used in different evaluation campaigns. Well known automatic evaluation metrics are used to compare the suggested methods to a state-of-the art baseline and results are analysed and discussed.

Acknowledgements

I would like to take this opportunity to thank those who supported me in several ways.

First, I offer my sincerest gratitude to my former supervisor Christof Monz who introduced me to the field of natural language processing and supported throughout the time this thesis was created. I also thank my supervisor Thomas Roelleke who without his comments and constructive feedback this work would have been impossible to finish.

I would like to show my gratitude to my former and current colleagues at IR research group in Queen Mary, University of London for providing a stimulating research environment. My special thanks also to my friend Yasaman Soltan-Zadeh for many productive discussions and her invaluable comments.

Finally, I would like to thank my parents for their love and support during all these years.

Sirvan Yahyaei

London, October 2011

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Machine Translation[1] is the task of translating text or speech from one natural language to another by means of a computer software. Machine translation has been a challenging problem in artificial intelligence for decades. Several approaches have been researched and investigated since 1950s, however because of translation's difficult nature, the efforts had not been very successful until recently. In early 1990s, researchers at IBM started an approach based on information theory called **Statistical Machine Translation (SMT)**. Statistical machine translation or SMT tries to perform translation by using statistical methods and learning how to translate based on previously manually translated texts. The state-of-the art SMT is a supervised machine learning approach to translate sentences from *source* to *target* by using a bilingual parallel corpus of source to target sentences.

An ideal statistical machine translation system for sentence translation is consisted of a sentence dictionary with all the possible sentences with their correct translations. However, apart from computational complexities of such a system, building such a system is not practical, because language vocabularies are not finite and contain open word classes. In addition, there is no limit for sentence length in practice. Therefore, SMT approaches need to employ a method to segment sentences into smaller units and maintain dictionaries to translate those units. The pioneering work in IBM, considered words as these units and translated sentences word by word. Later, phrase-based mod-

[1] Also called automatic translation or MT.

els which take a sequence of words as the basic unit were proposed that substantially improved the quality of translation over the previous word based model. In this research, we explore methods to improve the quality of phrase-based statistical machine translation and evaluate SMT quality beyond automatic metrics.

There are several factors that make MT in principle a difficult task. For a few examples: words in different languages do not always have a one-to-one relationship. Sometimes concepts are expressed differently in different languages. There are many ambiguous phrases and terms that need a very wide context to resolve. Additional to above examples, different languages differ in their syntactical structure and one of the important syntactical differences is word ordering. Even languages with similar main word order may not have the same word order in expressing the same concept. Recent studies show that word reordering accounts for a large portion of performance variability among European languages [Birch et al., 2008]. For an example of the effect of reordering on the output structure and also correct phrase translation, consider the example in Figure 1.1. This example shows a German sentence translated into English. The SMT decoder can not easily skip the distance between `will` and `erfahren` to correctly translate them into `wants to know`, because there are more than 18[2] words between the two phrases and considering all the permutations possible is intractable.

Current state-of-the art statistical machine translation approach is called phrase-based SMT. The basic unit for translation in PBSMT is a phrase or a sequence of words. There are different types of PBSMTs. Firstly, non-syntactic models that consider any sequence of words as a phrase ad generate the translation by composing target phrases. The source and target phrases for these models can be contiguous or with gaps that can be filled later by other phrases. The sequence of words does not need to be a syntactic unit. This characteristic gives the PBSMT systems the ability to learn a vast number of phrases including non-syntactic phrases that have been shown to be very useful in capturing obscure and complex phrase translations across different languages. Secondly, there are models which are based on phrases, but with restrictions on source or target sides to be syntactic units. The syntactic constraints prevents these models to benefit

[2] 18 non-tokenised words; if the sentence is tokenised, based on the method of tokenisation there will be more than 18 words.

DE	Der SPD-Haushaltsexperte Johannes Kahrs *will* von Kanzlerin Angela Merkel Einzelheiten über die Feier im Kanzleramt anlässlich des 60. Geburtstages von Deutsche-Bank-Chef Josef Ackermann *erfahren*.
MT	The SPD budget expert Johannes Kahrs *wishes* of Chancellor Angela Merkel in the Chancellery of details of the ceremony to mark the 60th Birthday of German Bank chief Josef Ackermann *learned*.
HUM	The SPD budget expert Johannes Kahrs *wants to know* from Chancellor Angela Merkel the details of the ceremony in the Chancellery to mark the 60th birthday of Deutsche Bank CEO Josef Ackermann.

Figure 1.1: A German sentence that requires a long distance reordering to correctly translate the verb. DE is the German sentence, MT is the output of the machine translation system and HUM is the human translation.

from the useful non-syntactic phrase, however allows them to generate more grammatically correct translation and easily take advantage of syntax in reordering decisions. In all the models and experiments described in this work, the former approaches are used and investigated.

The main focus of this research is the problem of reordering in phrase-based statistical machine translation. Because of the core role of reordering in machine translation, it affects several aspects of translation. On one hand, better reordering directly influences the quality of the output. On the other hand, the reordering method has a significant impact on determining the performance of the translation algorithm in terms of speed. In addition to improving and investigating reordering, we explore the problem of cross-lingual text fragment alignment, which is not directly affected by reordering. For this problem, the focus is the effectiveness of similarity measures based on divergence from randomness and word-based translation models without reordering constraints.

In the following chapters of this thesis, we present methods of performing reordering that lead to better translations, while investigating their effect on the speed of the main algorithm. In addition, we analyse the effect of reordering on the quality of translation for specific natural language tasks. In other words, apart from automatic evaluation of the algorithms and the contribution of the reordering models by general evaluation metrics, we evaluate the quality of the generated translations for specific applications such as named entity recognition. Further than that, word-based translation models are combined with similarity measures based on models of randomness to perform mono-lingual and cross-lingual text fragment alignment and analyse the effectiveness of the models in the aforementioned settings.

In this research, we aim to improve the quality of statistical machine translation with respect to automatic evaluation metrics by focusing on the reordering phenomenon. The models proposed in this work are independent from the source and target languages and solely rely on statistics collected from the bilingual parallel data. In addition, common natural language applications such as named entity recognition and text fragment alignment have been selected for evaluation to go beyond the general automatic metrics and explore the effectiveness of the models in improving translation quality for particular tasks. Likewise, the method to perform text fragment alignment is language independent and relies on word statistics in the documents. The word-based translation models are built using parallel corpus and similarity measures use DFR models and the word-based translation models to estimate the text fragment similarities.

Throughout this thesis all the models and methods have empirically been evaluated. Human evaluation of machine translation can be very expensive to perform and impractical in evaluating several methods repeatedly. We have used automatic evaluation metrics and test collections provided by international evaluation campaigns for the experiments. All the proposed approaches are compared to the state-of-the-art baselines, trained and tuned on the same data. The baselines are all well-established approaches in the community with strong performances at the time of experiments. Even though, the benchmarking experiments can not fully show that our approaches are always and on all data superior to other methods, they are providing a level field for comparing the proposed models with established ones. Additionally, we have tried to perform the experiments across different language pairs and several test sets to make the experiments representative of real world applications.

1.1 Research Questions

1. How can one take advantage of the fact that words tend to move together when they are translated across languages?

2. Is chunking and grouping words together a helpful solution for long-distance reordering?

3. How important and effective is language modelling in dealing with the reordering problem?

4. Are distortion constraints influence the quality of translation significantly and how can an important parameter such as distortion limit be tuned to avoid using the same parameter for sentences with different structure?

5. What kind of features in a reordering model help to relax the reordering constraints[3] in phrase-based SMT without degrading the performance of the algorithm in terms of speed and quality?

6. Does adjusting the distortion limit improve quality of the translation compared to manual tuning?

7. What is the effect of being cross-lingual on text fragment alignment and is the difference between the performance of the mono-lingual algorithm and cross-lingual algorithm substantial enough to rule out the full translation as a viable approach in performing fragment alignment?

8. Is translating to English and using available tools in English to perform NLP tasks such as named entity recognition a viable alternative to multi-lingual tools?

9. What is the effect of improving reordering on different NLP tasks for different language pairs? and is improving the reordering going to improve the quality of these tasks for all language pairs?

[3] such as distortion limit (see Chapter 2, Section 2.5)

1.2 Contributions

We present an approach to extend the phrase-based decoder that **dynamically chunks, reorders, and applies phrase translations in tandem**. By grouping words and moving them together, we try to enable the decoder to consider long-distance re-orderings and avoid unnecessary short distance permutations. In addition, our method does not rely on language-dependent parsers or chunkers and uses the word alignment information to build the chunker. To keep the search space manageable, phrases inside the chunks are monotonically translated, thus by eliminating the unnecessary local re-orderings, it is possible to perform long-distance re-orderings beyond the common fixed distortion limit.

To overcome the issue of setting the optimum distortion parameters in the phrase-based decoders and the fact that different sentences have different reordering requirements, **a method to predict the necessary distortion limit for each sentence and each hypothesis expansion** is proposed. A discriminative reordering model is built for that purpose and also integrated into the decoder as an extra feature. Many lexicalised and syntactic features of the source sentences are employed to predict the next reordering move of the decoder. The model scores each reordering before the sentence translation, so the optimum distortion limit can be estimated based on these score.

We devise **a method to evaluate the performance of state-of-the-art named entity recognisers on the text translated by a SMT**. Specifically, we investigate the effect of word reordering and incorporating reordering models in improving the quality of named entity extraction.

Finally, we propose **an approach to automatically align fragments of texts of two documents in different languages**. A text fragment is a list of continuous sentences and an aligned pair of fragments consists of two fragments in two documents, which are content-wise related. Cross-lingual similarity between fragments of texts is estimated based on models of divergence from randomness. A set of aligned fragments based on the similarity scores are selected to provide an alignment between sections of the two documents.

1.3 Thesis Outline

The thesis is structured as follows:

- Chapter 2 provides the background on statistical machine translation needed for the rest of the thesis. It gives a brief overview of word-based models and introduces the main concepts of phrase-based models and their variations. We also describe some of the main evaluation metrics currently used in the machine translation community and the discussions on their effectiveness.

- Chapter 3 discusses the reordering phenomenon in machine translation. We present a description of the problem and various reordering requirements of different language and overview the previous and current approaches proposed to deal with reordering so far.

- Chapter 4 introduces an approach of dynamically chunking and translating in tandem. The proposed method enables the decoder to consider permutations which include long distance re-orderings. Several examples are shown to demonstrate that by grouping words and moving them together, the decoder is able to consider long-distance re-orderings and avoid unnecessary short distance permutations.

- Chapter 5 presents a new method that aims to dynamically adjust the distortion limit in phrase-based decoding. Adjusting the distortion limit prevents the decoder to explore undesirable parts of the search space. The performance of this approach is compared to several other systems in an evaluation campaign and the results are discussed in this chapter.

- Chapter 6 describes an approach to evaluate the models and MT in general in the context of several NLP tasks. Since the quality of even the best SMT systems differs for different language pairs and heavily depends on the available language resources, it is important to evaluate the performance of different NLP tasks on machine translation output.

- Chapter 7 presents an approach to align text fragments of two documents in two different languages. One aim of the chapter is to investigate the effectiveness of divergence of randomness model in the context of cross-lingual fragment alignment.

- Chapter 8 concludes the work done in the thesis and proposes some directions for future research.

1.4 Publications

The work presented in Chapter 4 is also described in *"Decoding by dynamic chunking for statistical machine translation"* presented at MT Summit XII [Yahyaei and Monz, 2009]. Chapter 5 was presented as *"Dynamic distortion in a discriminative reordering model for statistical machine translation"* in the seventh International Workshop on Spoken Language Translation (IWSLT) [Yahyaei and Monz, 2009]. Chapter 7 is accepted as *"Cross-lingual text fragment alignment using divergence from randomness"* to be presented in the 18th edition of the International Symposium on String Processing and Information Retrieval (SPIRE) [Yahyaei et al., 2011], which is a joint work with Marco Bonzanini.

CHAPTER 2

Statistical Machine Translation

The subject of this thesis it to explore and examine methods and models to improve and evaluate reordering in statistical machine translation. In this chapter, we briefly introduce the basic foundations of machine translation and specifically discuss statistical MT concepts and methods used throughout the rest of the thesis. Since there are several different approaches to machine translation and even various views in statistical MT, we do not touch upon all the methods and approaches in this chapter. The main aim of this chapter is to clearly introduce the terminology used in this work. First, we give an introduction to statistical machine translation and various stages used in training the models. Then we discuss the procedure of evaluation in machine translation and describe the most common used automatic metrics available today. Finally, we explain the in-house decoder developed during this research and its main components.

2.1 Machine Translation

Machine translation is the task of translating text or speech from one natural language to another by means of a computer software. Due to the complex nature of natural language, there are many challenges in statistical machine translation. For example, lexical ambiguity which occurs due to the limitation of fully modelling of the context; different languages have different word orders; in many cases, syntactic structures are not preserved across translation; treatments of tenses are different in different languages and so on. There have been several approaches to perform machine translation since the beginning of computers. Recently the use of statistical methods in translation, as well as natural language processing tasks, has been very successful.

Statistical machine translation tries to perform translation by using statistical methods. The process is mapping sentences from the **source** language into sentences from the **target** language. The main idea of statistical machine translation is automatically translating by means of models estimated from parallel and mono-lingual corpora. A parallel corpus is a set of source-target documents that are translation of each other. Huge amount of text in different languages and the existence of massive computational power and distributed algorithms have made SMT a very strong candidate in the MT industry[1].

2.2 Statistical Machine Translation

Although the idea of using statistical methods was first suggested in 1949 [Weaver, 1955], the current SMT approaches are based on the pioneering work started in early 1990s in IBM [Brown et al., 1990, 1993]. The initial models by IBM, were called word-based models. In word-based models, translation units are words and the selection of translation options is mainly done by a combination of translation probabilities and the fluency of the generated output.

Later, phrase-based models which consider a sequence of words as the basic unit, were proposed [Och et al., 1999; Koehn et al., 2003] that substantially improved the quality of translation. The improvement was achieved by automatically taking into account the local context for many of replacements and implicitly addressing many of the local reorderings. Here, we formally define the machine translation problem and introduce the main methods of solving it.

Assume we want to translate a foreign sentence $\mathbf{f} = f_1^J = f_1, ..., f_J$ into a target sentence $\mathbf{e} = e_1^I = e_1, ..., e_I$. The problem of statistical machine translation can be written as the following equation:

[1] At the time of writing, Google translate provides a free online service for translation between 63 languages (`http://translate.google.com`)

$$\hat{\mathbf{e}}(\mathbf{f}) = \arg\max_{\mathbf{e}} \{Pr(\mathbf{e}|\mathbf{f})\} \tag{2.2.1}$$

where arg max is the search problem for finding the target sentence. There are many approaches to solve the equation 2.2.1: Syntax-based methods that deal with the problem as a tree-to-tree [Yamada and Knight, 2001], tree-to-string or string-to-tree [Galley et al., 2004] mapping. Phrase-based models such as [Koehn et al., 2003; Och and Ney, 2004] use aggressive methods to learn contiguous phrases from the parallel corpus and use them during translation. A modified version of phrase-based models is hierarchical phrase-based model which learns both contiguous and non-contiguous phrases and differently use them for translating [Chiang, 2005, 2007].

Independent of general approach to the search problem, they all mainly rely on two models called *translation model* and *language model*. A translation model consists of two elements. Firstly, a series of *rules* that describe the steps to transform a source sentence to a target sentence [Lopez, 2009]. Secondly, a set of *parameters* that are used to score the unweighted ruleset, mentioned before. The process of assigning values to these parameters is called *parameter estimation* [Lopez, 2008]. A language model is an order n Markov chain model that assigns a probability to each string. Languages models are built for the target language to score the produced hypothesis at each step. We will discuss each model further later in this chapter.

Apart from the two main models mentioned above, many other models and penalties have been suggested to guide the search process. All these models are integrated in a uniform way as features and their importance is determined though empirical means.

Equation 2.2.1 can be rewritten as:

$$\hat{\mathbf{e}}(\mathbf{f}) = \arg\max_{\mathbf{e}} (\Sigma_i \lambda_i f_i(\mathbf{e}, \mathbf{f})) \tag{2.2.2}$$

where f_i is a feature and λ_i is its weight in this log-linear model. Current SMT models employ a wide range of features to perform the task of translation. For an example of a set of features, the following list is a standard list used in many baseline phrase-based systems:

- phrase translation probabilities and lexical probabilities for both directions.
- an n-gram language model.
- phrase and word penalties.
- distance-based reordering penalty.

2.3 Translation Model

To build translation models, we need to extract a ruleset, which as defined before is a set of mapping from the source strings to the the target strings. In word-based models, the ruleset is basically a dictionary of source words to target words. After extracting the ruleset, a weight function must be defined to score each rule and assign a weight or several weights to each rule.

2.3.1 Word-Based Models

The word-based models, as a first statistical approaches to machine translation, were introduced in the late of 1980s and early 1990s by IBM [Brown et al., 1990, 1993]. Since it is difficult to manually align enough sentences to be used for learning the probabilities for each language pair, the word alignment process starts with non-aligned parallel sentences and most of the times an unsupervised machine learning algorithm such *Expectation Maximisation* [Dempster et al., 1977] to iteratively learn the probabilities.

In the IBM models, Model 1 is the simplest model and later models extend and improve on it. Model 1 is relatively easy to integrate into the EM algorithm and is fast to train. The generative story of IBM Model 1 goes like this: given an English sentence, choose a length for a French sentence. Then for each position in the French sentence uniformly connect it to an English position and decide what French word to be there. All the connections in this model are equally likely and order of the words is irrelevant. Since

Model 1 is an easy and fast model to train, it is a good start to provide initial estimations for higher models.

In Model 2, the ordering and position of the words relative to the length of the sentence is added. The HMM Model of [Vogel et al., 1996] is not an IBM model, but it is widely used for word-alignment in a mixture with IBM models. This model adds a relative ordering model compared to the absolute model of Model 2. Model 3 adds fertility probabilities that give the probability of how many French words are generated by an English word. Model 4 adds a relative reordering model and Model 5 fixes the deficiency of models 3 and 4.

Although the word-based models are not state of the art any more, most of the concepts and methods are still used. The above word-based models are widely employed to produce word alignments that are used to make learning phrases practical in phrase-based models.

Figure 2.1 shows a visualisation of a word alignment between a German sentence and its translation in English. The alignments between words are shown by black cells in the matrix. As you can see, words may have multiple or no alignment points. Unfortunately, finding the perfect alignment between a sentence and its translation is not always possible. Sometimes, function words do not have a clear correspondence in the other language, but still make changes to the other words. Another source of problem is words inside idioms that can have a completely different equivalent outside of the idiom.

A common approach to improve the alignment quality is called symmetrisation of word alignments, which is finding the alignments in both directions and then using a method such as intersection or union to combine both alignments [Och and Ney, 2003, 2004]. Figure 2.1 shows the result of the symmetrisation process after taking the union, while Figure 2.2 shows the same sentence pair and the word alignment produced by the intersection. It is clear from the figures that the number of aligned points are less in the `intersection` method, which is mostly used to produce high precision word alignments. On the other hand, to achieve high recall the `union` method is preferable. A middle-ground approach which is commonly used in phrase-based models is a heuristic approach that starts with the `intersection` method and adds new alignment points based on a few criteria [Och et al., 1999; Koehn et al., 2003]. Figure 2.3 shows the previous sentence pair aligned by the heuristic approach called `grow-diag-final-and` [Koehn, 2009]. In `grow-diag-final-and`, the reliable alignment points of `intersection` are taken and some of the points produced by the `union` method are added. This approach is based on the observation that good alignment points are in the neighbourhood of other points. `diag` in `grow-diag-final-and` means that diagonal neighbours that are in `union`, but not in `intersection` are added. In the `final` step, alignment for still unaligned words are added, however, the `and` tag restricts this step to alignment points that both words are unaligned.

Apart from symmetrisation, there have been many suggestions to improve the word alignment quality [Cherry and Lin, 2003; Moore, 2004] and even completely different approaches to the original IBM models [Ittycheriah and Roukos, 2005; Moore, 2005], however some studies have shown that word alignment quality has a minor impact on the translation quality of phrase-based models [Ayan et al., 2005].

2.3.2 Phrase-Based Models

The next generation of statistical machine translation systems after word-based models is phrase-based models which translation units are multi-words or **phrases**. As mentioned before, there are several models based on phrases. A string-to-string model, that both source and target phrases are contiguous, are simply called phrase-based models [Koehn et al., 2003]. Another model that uses phrases with gaps that can be filled with other phrases is called hierarchical phrases-based model [Chiang, 2007]. In addition there are several models that constraint phrases of one or both sides to constitute a syntactic unit. These models are models are generally called syntax-based models [Yamada and Knight, 2001; Galley et al., 2004; Marcu et al., 2006].

Figure 2.4 shows a translation process in a phrase-based model[2]. One of the advantages of phrase-based models over word-based models is resolving the problem of multiple

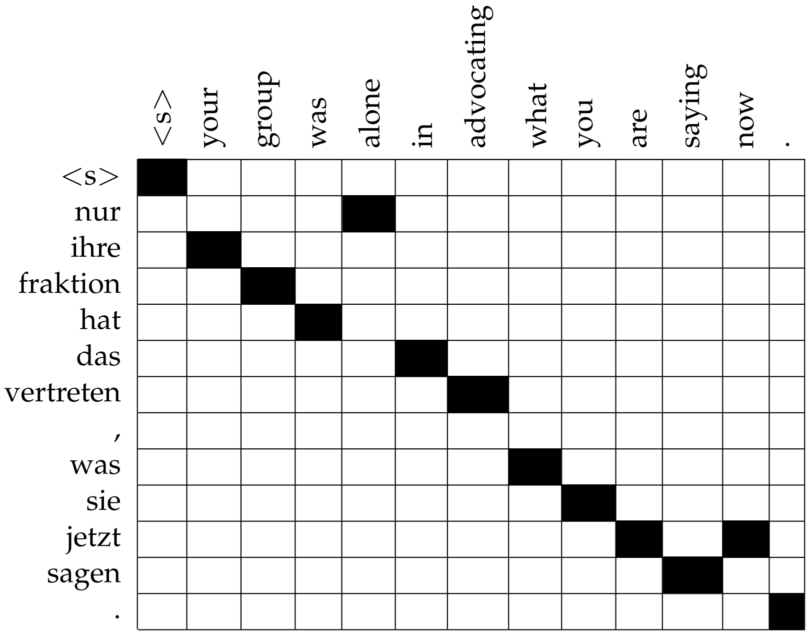
[2] This is an example of phrase-based models that is not hierarchical and phrases are contiguous strings without gaps

	<s>	your	group	was	alone	in	advocating	what	you	are	saying	now	.
<s>	■												
nur					■								
ihre		■											
fraktion			■										
hat				■									
das						■		■					
vertreten							■						
,													
was								■					
sie									■				
jetzt										■		■	
sagen											■		
.													■

Figure 2.1: Word alignment between a German sentence and an English sentence; symmetrisation is done by the `union` method.

	<s>	your	group	was	alone	in	advocating	what	you	are	saying	now	.
<s>	■												
nur					■								
ihre		■											
fraktion			■										
hat				■									
das													
vertreten							■						
,													
was								■					
sie									■				
jetzt												■	
sagen											■		
.													■

Figure 2.2: Word alignment between a German sentence and an English sentence; symmetrisation is done by the `intersection` method.



	<s>	your	group	was	alone	in	advocating	what	you	are	saying	now	.
<s>	■												
nur					■								
ihre		■											
fraktion			■										
hat				■									
das						■							
vertreten							■						
,													
was								■					
sie									■				
jetzt										■		■	
sagen											■		
.													■

Figure 2.3: Word alignment between a German sentence and an English sentence; symmetrisation is done by the `grow-diag-final-and` method.

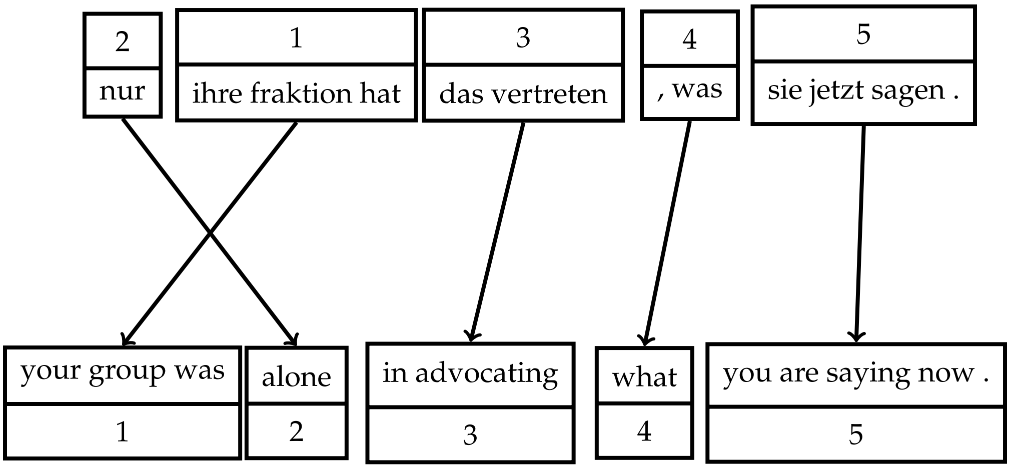


Figure 2.4: An example of translation by phrase-based models.

mappings. Since there are a lot of one-to-many or many-to-one mappings in translation, words as base units are not enough to resolve them. Additionally, translating a group of words together helps to incorporate more context and resolve many translation ambiguities. As we will discuss later, a big advantage of phrase-based models over word-based models is capturing the local re-orderings in phrase pairs.

As mentioned before different phrase extraction methods are available in the literature. [Marcu and Wong, 2002] present a joint SMT model to learn phrases from parallel corpus. However, despite its mathematical foundation, the search space is too large to be practical for current collections. Instead, here we explain a heuristic approach which is widely used in the community. Our description is based on [Koehn et al., 2003].

For each sentence pair, we collect all the phrases that are consistent with the word alignment of the sentence pair. A phrase pair $(\tilde{f}, \tilde{e})$ is consistent with alignment A, if all the words in $\tilde{f}$ and $\tilde{e}$ have alignment points with each other. In other words, we extract following set of phrases:

$$\begin{aligned}
\{(\tilde{e},\tilde{f})| \quad & \forall e_i \in \tilde{e} : (e_i, f_j) \in A \rightarrow f_j \in \tilde{f} \\
\wedge \quad & \forall f_j \in \tilde{f} : (e_i, f_j) \in A \rightarrow e_i \in \tilde{e} \\
\wedge \quad & \exists e_i \in \tilde{e}, f_j \in \tilde{f} : (e_i, f_j) \in A\}
\end{aligned}$$

where the last constraint ensures that, there is at least one alignment point between the words inside the phrase pair.

Figures 2.5, 2.6 and 2.7 show the extracted phrases from the sentence pair in Figures 2.1, 2.3 and 2.2 respectively. Since there are less alignment points in the `intersection` method, there are more extracted phrases compared to the other two word alignment methods. As you can see, there might be more than one mapping for some of the phrases or no mapping at all. Note that, this method of phrase extraction does not require any kind of syntactic structure from a string of words for being a phrase. Surprisingly, this is one of the main strengths of phrase-based models over syntax-based models. In Section 3.2.2, we compare these approaches on this feature.

nur ||| group ||| 0-0
nur ihre ||| group was ||| 0-0 1-1
nur ihre fraktion ||| group was alone ||| 0-0 1-1 2-2
ihre ||| was ||| 0-0
ihre fraktion ||| was alone ||| 0-0 1-1
fraktion ||| alone ||| 0-0
hat ||| your ||| 0-0
vertreten ||| advocating ||| 0-0
vertreten ||| advocating what ||| 0-0
was ||| are ||| 0-0
sie jetzt sagen ||| saying now ||| 0-0 1-1 2-0
jetzt ||| now ||| 0-0
. ||| . ||| 0-0

Figure 2.5: All the extracted phrases from a word alignment matrix produced by the `union` method symmetrisation in Figure 2.1. The third column is the inside phrase word alignment links.

das ||| in ||| 0-0
das vertreten ||| in advocating ||| 0-0 1-1
das vertreten ||| in advocating what ||| 0-0 1-1
vertreten , ||| advocating what you ||| 0-0 1-2
, ||| you ||| 0-0
, ||| what you ||| 0-1
, was ||| you are ||| 0-0 1-1
, was ||| what you are ||| 0-1 1-2

Figure 2.6: Extra phrases extracted from a word alignment matrix produced by the `grow-diag-final-and` method symmetrisation in Figure 2.3. Note that all the phrases extracted in Figure 2.5 is also extracted in this method.

```
ihre fraktion ||| was alone in ||| 0-0 1-1
das vertreten ||| advocating what ||| 1-0
das vertreten ||| in advocating ||| 1-1
das vertreten , ||| advocating what you ||| 1-0 2-2
vertreten ||| in advocating ||| 0-1
vertreten ||| in advocating what ||| 0-1
jetzt sagen ||| saying now ||| 0-1 1-0
jetzt sagen . ||| saying now . ||| 0-1 1-0 2-2
sagen ||| saying ||| 0-0
```

Figure 2.7: A few examples of extra phrases extracted from a word alignment matrix produced by the `intersection` method symmetrisation in Figure 2.2. These are phrases in addition to the phrases extracted in 2.5 and 2.6.

To estimate a probability for each pair, we simply use relative frequency:

$$\phi(\tilde{f}|\tilde{e}) = \frac{\text{count}(\tilde{e},\tilde{f})}{\Sigma_{\tilde{f}_i}\,\text{count}(\tilde{e},\tilde{f})} \tag{2.3.1}$$

In addition to relative frequency probabilities, many smoothing methods have been proposed to overcome the problem of sparse data [Zens and Ney, 2004; Foster et al., 2006].

Despite their success over word-based models, phrase-based models have some limitations. Firstly, according to the phrase extractor algorithm non-contiguous phrases can not be extracted. For example, in the German sentence, "Ich habe das Haus gekauft" and English sentence "I bought the house", a very good phrase is (habe...gekauft, bought). However, the phrase extraction algorithm is not able to capture the phrase without including (das Haus, the house) pair. An alternative approach to address this issue is proposed in [Chiang, 2005] which will be discussed in Section 3.2.4. Secondly, as it will be discussed in detail later, despite their effectiveness in reordering of the words inside phrases, phrase-based models are not very good in capturing the reordering requirements between phrases.

2.4 Language Model

An important component of each statistical machine translation system is the *language model*. Generally, a language model estimates how likely is a sentence or a sequence of words to be uttered by a native speaker. A statistical language model gives a probability to a sequence of words based on the context of sequence. The most common type of language model is n-gram language models. An n-gram language model is a Markov model of order n which gives the probability of seeing a given word only based on the last $n-1$ words preceding it [Manning and Schtze, 1999]:

$$P(w_1^k) := \Pi_1^k P(w_i|w_{i-n+1}^{i-1}) \tag{2.4.1}$$

where w_1^k represents a sequence of k words $w_1, w_2, \cdots, w_k$. N-gram language models are very effective in word selection of tasks such as Automatic Speech Recognition [Bahl et al., 1990]. For an example a good language model should give "*this is a small house*" a higher probability than "*this is a small home*". This example shows, how a language model feature aids a statistical machine translation system in choosing words in different contexts. Lexical probability feature might give "*home*" a higher probability than "*house*", but the language model steps in to select the better overall hypothesis. Another area that language models help statistical machine translation is word reordering. For example, suppose we have two hypotheses "*this is a house*" and "*this a is house*". A language model must give the first sentence a higher probability, so the decoder prefers the hypothesis with the correct word order over the other one. In general, a language model feature selects a reordering choice only if it leads to a translation that seems to be a better sentence than the other alternatives. In other words, for a language model feature the source sentence is irrelevant and the origin of the reordering is not taken into account. It only scores the generated target hypotheses and favours the ones that are more likely.

In Chapter 3 we argue that despite the relative effectiveness of language models in selecting words and in local reorderings, they are not sufficient to address the general problem of reordering.

2.5 Decoding

Recalling Figure 2.4, to translate a sentence, apart from finding equivalent phrases from the phrase table. we need to reorder the phrases to build a grammatically correct target sentence. Due to the problems of the alignment process and aggressive nature of phrase extraction algorithm, there are many, usually more than 30, candidates for each phrase. Considering the number of possible permutations even for a short sentence and possible translations for each phrase, the search space is overwhelmingly big. [Knight, 1999] shows that searching among all possible translation options is an NP-complete problem. The state-of-the-art SMT systems employ a set of features to model different aspects of the translation problem and use a dynamic programming approach to explore a part of search space and maximise the right hand side of equation 2.2.1. One way to limit the number of translation options is to constraint the window size that words can permute in. However, still the search space is large enough to make the translation process on modern machines impractical. A widely used technique for decoding in statistical machine translation systems is *Beam Search*.

To translate a sentence with beam search decoding, we try to find a chunk of words, a phrase, that we know how to translate, which means we have at least one equivalent phrase in our phrase table for it. After generating a hypothesis for each equivalent phrase of this phrase, the next phrase is selected to translate. Meanwhile, to enable reordering of the phrases, the next phrase is selected from a window with a specific length [3], around the previous phrase. Continuing this approach, each hypothesis is expanded by selecting and translating a next phrase. In beam search technique, we want to keep the best partial translation and discard those which are worse than a threshold. According to equation 2.2.2 we can compute the score for each hypothesis to compare to others and discard some of them, which means, we do not expand them any more. Since while we translate more, the score of hypotheses get lower, we organise the hypotheses based on the number of the words that have been translated so far, so we can compare the candidates that have done a same amount of job.

[3] distortion limit

2.6 Evaluation

How we can measure the quality of a statistical machine translation system is a very crucial question. An intuitive method to evaluate machine translation output is manual evaluation where human annotators understand at least the target language and evaluate the output in **adequacy** and **fluency**. Obviously, due to huge amount of human effort requirement and also disagreement between annotators, even inconsistency of the judgements of one annotator over time, this is not a practical approach to evaluate a system. For example, during development of a system we need to evaluate the output several times a day, therefore having an automatic, easy to use and cheap evaluation method is essential.

A good evaluation measure should be fast, automatic and consistent. In addition, we need a measure that correlates well with human judgement. Some evaluation measures are borrowed from Automatic Speech Recognition such as Word Error Rate [McCowan et al., 2004], however because they have been designed for another task, they do not measure all the aspects of the translation quality and do not correlate well with human judgements.

Many evaluation measures have been proposed in the SMT community. Here we briefly introduce the most important ones:

- **BLEU** It measures the precision of uni-gram, bi-gram, tri-gram and four-gram of the output with respect to one or more reference translations. Additionally, it has a penalty for short sentences. BLEU measures the accuracy, so higher BLEU scores are better. The BLEU-4 metric is defined as:

$$BLEU\text{-}4 = brevity\text{-}penalty \prod_{i=1}^{4} precision_i \tag{2.6.1}$$

  where *brevity-penalty*, as we said, is used to penalise dropping words and generating short sentences. In other words, it is simply measuring the recall. $precision_n$

is the n-gram precision which is:

$$precision_n = \frac{\text{number of correct } n\text{-grams}}{\text{number of } n\text{-grams in the reference sentence}} \tag{2.6.2}$$

[Papineni et al., 2001]

- **NIST** Very similar to BLEU, it is a weighted n-gram precision with penalty for short sentences [Doddington, 2002].

- **METEOR** In this metric evaluation is done through a sequence of stages, which in each stage a set of matching uni-grams will be found and scored. For example, the first stage is the exact matches and the second is the match of stemmed words. METEOR is based on the weighted harmonic mean of Precision and Recall [Banerjee and Lavie, 2005; Lavie and Agarwal, 2007].

- **TER** This metric measures the number of edits required to transform an output into one of the translation references. Edits include **insertions**, **deletions**, **substitutions** and **shifts**. Also, capitalisation and punctuation errors can be included. TER is equal to the number of above edits divided by the average number of reference words, where the main reference, which edit operations are calculated against it, is the closest one to the output [Snover et al., 2006].

Although, there are many debates about the best evaluation measure in machine translation community [Callison-Burch et al., 2006], BLEU is currently widely used and almost all researchers report their experiments based on the BLEU metric. The state of the art SMT system achieves BLEU scores in different ranges for different language pairs. Some language pairs are more difficult and the BLEU scores are lower compared to others. In addition, the number of reference translations affects the range of the BLEU score achieved by the state of the art systems. For example in WMT 2011, the best performing system achieved the BLEU score of 0.25 for the German to English task and 0.17 for the English to German task [Callison-Burch et al., 2011]. On the other hand in this thesis, we report experiments on Arabic to English with BLEU scores close 0.60. The test data set for Arabic to English has 16 reference translation compared to 1 reference translation for the German to English data set. Although the automatic metrics are designed for comparing systems and most of the times the absolute values do not have specific meanings, a 0.1 BLEU points improvements is very likely to mean noticeable positive changes in the translation quality.

To evaluate the translation systems, apart from the training data which is used to build the translation and language models, we generally use two different sets, including source sentences and their references, namely development and test sets. The development set is used to tune the parameters of the system, particularly the weights of the features in the log-linear model. The test set is the set that the final score is reported and is used to in comparison of the systems. Usually, a test or development set contain 1000 sentence with 4 reference translation for each sentence, and if 4 reference translations are not available, a 2000-sentence set is used to make the result of the tuning and testing reliable.

2.7 TAGINE: Phrase-based Decoder

Similar to other natural language processing tasks, in statistical machine translation, to empirically verify every new approach or idea to get accepted by the community as an effective approach, it needs to be compared to valid baseline. Therefore, along the literature study we implemented a state of the art phrase-based decoder as a foundation for our future experiments. We re-implemented the existing phrase-based decoder in the group with focusing on modularity and speed. To reach a good speed performance and also taking advantage of object orientation for modularity, we developed the system in Java and C++ programming language.

Tagine[4], is a phrase-based multi-stack, multi-beam decoder with ability to add more features very easily. Main features are categorised in three classes: Translation model, language model and distortion model. Additional features such as phrase penalty [5] can be integrated without affecting other components of the system. All the features are managed by a component which acts as an interface between the features and the

[4] TrAnslation enGINE

[5] A simple feature that encourages applying longer phrases during decoding.

main decoder. Tagine has a component to manage all the constants and parameters of the decoder through configuration files, so many adjustments can be made without recompiling the system. It reads a generated phrase table in a widely used format as its translation model and has an interface to interact with SRILM [Stolcke, 2002] and IRSTLM [Federico and Cettolo, 2007] language model tool-kits.

We have developed a web application which connects to Tagine through a socket and can perform the translation online. This version has an implementation of the binary phrase table of [Zens and Ney, 2007], so it can use very large phrase tables or multiple languages simultaneously.

As described in Section 2.2 the problem of statistical machine translation can be written as:

$$\hat{\mathbf{e}}(\mathbf{f}) = \arg\max_{\mathbf{e}}(\Sigma_i \lambda_i f_i(\mathbf{e}, \mathbf{f})) \tag{2.7.1}$$

where f_i is a feature and λ_i is its weight in this log-linear model. The performance of the this translation system is largely dependent on the finding proper weights for the features (λs). To optimise the weights, we use Minimum Error Rate Training algorithm of Franz Och [Och, 2003]. To find the best weights for the features, MERT algorithm needs a large amount of candidate translations, so it iteratively runs the decoder with the best weights from the previous iteration and cumulatively use the translations to find the minimum points.

Summary

In this chapter, we briefly introduced machine translation and the main ideas of statistical MT. The word-based models of IBM and their application in the heuristic phrase extraction method of phrase-based models were also discussed. We pointed out that the two main models of statistical MT are translation and language models and described some of their attributes. In addition in Section 2.6, some of the main automatic metrics and the logic behind MT evaluation were discussed. This chapter was concluded by a description of our implementation of a phrase-based decoder and experimental framework used for the rest of the thesis.

CHAPTER 3

Reordering in Statistical Machine Translation

There are several factors contributing to the difficulty of machine translation. For example, different levels of morphology between the source and target languages can make it difficult to generate the right verb, with correct morphology, in the target side. Another problem is the issue of unknown words. If both languages are using the same alphabet and are historically related, there is a chance of success in translating unknown words, if simply the unknown word is reproduced in the target side. However, if the languages are historically distant or more importantly the alphabets are different, this approach has a little chance to succeed.

One of the major problems in MT is different word orders between the source and the target language. In translating a source sentence to a target sentence, reordering is the requirement of the decoding algorithm to skip words and cover them after translating later words in the sentence. In other words, reordering in statistical machine translation is the need for not necessarily sequentially translate all the phrases in a source sentence.

The main reason that reordering is needed is the fundamental word order differences between languages. The main word order in some languages is Subject-Object-Verb such as Persian[1] and in some it is Subject-Verb-Object such as English. There are other word order differences, such as the order of noun modifiers and the noun between

[1] Although most of the sentences are in the form of SOV, the word order is almost free and many sentences can be written in different order while pertaining the same meaning.

different languages.

In this chapter, we define different types of reordering and overview the current approaches proposed to deal with the problem. In the rest of this chapter, we discuss the difference between the main SMT approaches in tackling the word order differences and the role of syntactic knowledge in some of these works.

3.1 Different Types of Reordering

We categorise different reorderings based on the width of the distance between the words that are moved to be translated consequently. There are many ways to categorise them into a few classes such as short, medium and long distance reorderings. Here, we discuss short and long distance reorderings and show examples of both to demonstrate the difference between them and the reason different methods are used to deal with them.

3.1.1 Short Distance Reordering

An intuitive difference in word order between languages is the location of noun modifiers with respect to the noun. For example, in English adjectives precede the noun, but in French they follow the noun. Figure 3.1 shows an alignment for a translation from French to English[2].

As you can see from the alignment matrix, to translate such a sentence, the decoder should allow a jump over the word *group* and a jump back to translate *pse*. This kind of permutation that requires a skip of a few words, less than three, is called short distance or local reordering. Fortunately, phrase-based models are relatively successful in this kind of reordering. Three different features of phrase-based models enable it to handle this situation effectively. Firstly, it is likely to extract the phrase *groupe pse* during training and translate this phrase in one phrase application, hence with correct word order. Secondly, *n*-gram language models are very effective to assign a higher probability to the correct permutation of the words in such a short distance. Finally, the distance

[2] $< s >$ in some of the alignment figures represents the beginning of the sentence.

	<s>	(	applause	from	the	pse	group	)
<s>	■							
(		■						
applaudissements			■					
du				■	■			
groupe							■	
pse						■		
)								■

Figure 3.1: A French to English translation with a local reordering.

based reordering model (see Section 3.2.1) does not penalise short distance jumps such as 1 and −1 too harsh, consequently, there is chance for other features like language model to compensate the penalty.

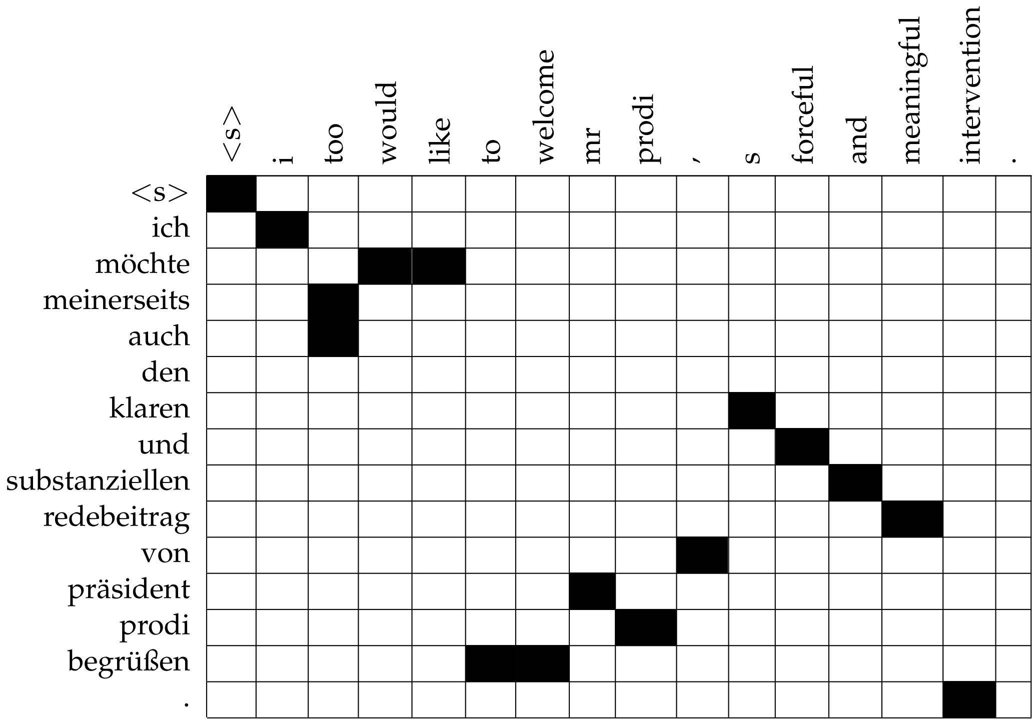
3.1.2 Long Distance Reordering

Although, a good language model or a lexicalised distortion model (see Section 3.2.3), can deal with local reordering problem, they are not usually sufficient for long distance reorderings. A long distance reordering[3] is needed, where one is dealing with languages with very essential different word orders. For example, many German sentences are in SOV order which is very different from English word order, SVO. In these cases, the problem is a long distance between the subject and the verb. After translating the subject, the decoder has to skip rather a large number of words to reach the verb and consequently jump back after translating the verb. Figure 3.2 shows an example of German to English translation which requires a long distance jump to correctly translating the verb.

3.2 Current Approaches to Tackle Reordering

In general, there are two main approaches to SMT: non-syntactic phrase-based and syntax-based. In both approaches there are different strategies to address the issue of

[3] Also called global reordering



	<s>	i	too	would	like	to	welcome	mr	prodi	,	s	forceful	and	meaningful	intervention	.
<s>	■															
ich		■														
möchte				■	■											
meinerseits			■													
auch			■													
den																
klaren											■					
und												■				
substanziellen													■			
redebeitrag														■		
von										■						
präsident								■								
prodi									■							
begrüßen						■	■									
.															■	

Figure 3.2: A German to English sentence with a very big difference in word order.

reordering. Non-syntactic phrase-based models, which are simply called phrase-based models, are divided into two classes: hierarchical and non-hierarchical or string-to-string phrase-based models.

Recently, there have been many efforts to incorporate syntax in statistical machine translation, particularly in word reordering. In syntax-based models the reordering problem is principally addressed by learning a ruleset that is constrained by syntactic rules. Generating a syntactically correct output with combining the phrase rules from the ruleset leads to implicitly performing the required reordering moves. Later in this chapter, we overview some of the ways of integrating syntactic information into SMT systems.

On the other hand, in phrase-based models there is no syntactic restrictions on phrases and reordering is mostly addressed by lexical models and features. Most of the phrase-based models rely on n-gram language models for their reordering decisions. As we will see later, the commonly used distance-based reordering model, prefers monotone translation over reorderings, unless there is enough evidence by the language model to show that the reordering is the case. As n-gram language model only considers the last $n-1$ words as context, they are not effective for distances longer than n. Another issue with current phrase-based models is distortion limit. To control the size of the search space, most phrase-based models, limit the length of the window that words can be reordered in. For similar languages in word order, this might be harmless, but for languages with totally different word order such as SVO and SOV languages, this limitation makes the finding of proper word order almost impossible. In summary, current phrase-based systems have a relatively limited ability to capture the word order differences between languages and require extra models to guide the decoder in making reordering decisions.

3.2.1 Basic Distance-based Penalties

IBM Distortion Models

A very basic and simple reordering model is first introduced by [Berger et al., 1996]. This models completely relies on the language model to select among the reordering options. A constant k is defined and the decoder is allowed to skip up to k words to translate the next word or phrase. [Zens et al., 2004] provide an overview of the effect of this simple reordering model on translation quality compared to monotone decoding.

Distance-based Reordering Model

Similar to the IBM distortion model, distance-based reordering model considers reordering relative to the previous phrase. We define the start position of the next phrase (the phrase we are about to translate) as $start_i$ and end_{i-1} as the end position of the previous phrase and i is the index of corresponding English phrase. So, the distortion cost of translating ith English phrase after $(i-1)$th, is:

$$d(i) = start_i - end_{i-1} - 1 \tag{3.2.1}$$

This model penalises more as the skips increases. If a phrase is translated exactly after the previous phrase, then the distortion cost is 0, because $start_i = end_{i-1} + 1$. Although, it is not necessary, we can convert $d(x)$ to a probability distribution by rewriting it as $d(x) = \alpha^{|x|}$ where $\alpha \in [0, 1]$ [Koehn, 2009]. Despite its simplicity, distance-based reordering is widely used in many baseline systems. The publicly available SMT system pharaoh [Koehn, 2004] uses distance-based reordering model and it is also the default reordering model of the open source SMT system, Moses [Koehn et al., 2007].

3.2.2 Syntax-based Approaches

Syntax-based SMT

Recently, many SMT systems started to incorporate syntactic information to capture the word order differences between the languages. [Yamada and Knight, 2001] allow reordering operations on syntactic parse-trees of the source sentence. Their model transforms a source parse-tree into a target string by applying learnt rules on the nodes of the tree. They define three operations, *reordering*, *inserting* extra words and *translating* leaf nodes, to transform the tree to a target string. The operations are applied based on probabilities learnt from the training data. Their syntax-based translation model was tested on translation from English to Chinese. Although, their SMT system were outperformed by phrase-based decoders [Koehn et al., 2003; Och and Ney, 2004], new generation of syntax-based decoders such as [Marcu et al., 2006; Galley et al., 2006] perform very well and in some cases better than phrase-based systems. [Galley et al., 2004] propose a linear algorithm to define a minimal set of syntax-based translation rules from word alignments. They explain a method to extract complex rules in a way to address the problems raised by [Fox, 2002]. In contrast to [Yamada and Knight, 2001] in [Galley et al., 2004] parse-trees of the target sentences are generated and reordering operations are taken place by extracted rules from the training data. [Galley et al., 2004] prepared a background theory to build a system which is able to properly explain the data and derive sufficient rules to perform the translation. In [Galley et al., 2006], they extend the framework to acquire not only a minimal set of rules, but a large number of more contextually richer rules. In addition, a method to estimate probabilities from training data is developed. Despite the success of the approach in modelling long distance reorderings with means of syntactic information, incapability of the approach to extract as much phrase rules as a phrase-based model do, leads it to a lower performance in some of the experiments.

[DeNeefe et al., 2007] compare the number of extracted phrases by syntax-based models and phrase-based models and try to find the reason of the issue of phrasal coverage in syntax-based models. With some modifications in [Galley et al., 2006] approach such as [Wang et al., 2007b] and combining rules extracted by another approach [Marcu et al., 2006], they report significant improvement over the baseline phrase-based system.

[Xiong et al., 2008] in a different approach integrated linguistic knowledge of syntactic and non-syntactic phrases into a BTG-based SMT system [Wu, 1997]. A maximum entropy based reordering model is built based on the lexicalised and syntax-based information of the phrases, to determine the order of the phrases in the BTG decoder, which are *inverted* and *straight*.

Reordering of the Source Sentence

In some approaches that have tried to employ syntactic information, transformation rules are applied to the source sentence to make it in an order similar to the target language. Transformation rules can be general syntax-based or specific lexicalised rules. Usually, in these approaches, source sentences of the training set are transformed and the reordered versions are used to learn the word alignments and phrases. [Xia and McCord, 2004] proposed a method to learn transformation rules from a parallel corpora. In their work, an algorithm is designed to extract re-write patterns, apply them to the source sentence and monotonically carry out the translation. At training time, to learn the rewrite patterns, source sentences are parsed, phrases are aligned and lexicalised and unlexicalised patterns are extracted. A simple rewrite pattern for reordering the *adjective*, *noun* phrases in English to *noun*, *adjective* phrases in French is as follows:

$$(NP \longrightarrow Adj \quad N) \Longrightarrow (NP \longrightarrow N \quad Adj)$$

where NP represents a *noun phrase*. Because there are many conflicting patterns during extraction, an organising and filtering method is applied to them. Also, a probability score is assigned to each pattern based on its count in the training data. Patterns in a group are sorted from the most specific patterns to the more general ones and are applied sequentially. Same pattern application is used to transform the source sentences of the test set and a phrase-based decoder is employed to monotonically translate them. Although, they have reported 10 percent improvement over the baseline phrase-based model, it is not mentioned in their paper the type of the reordering model in the baseline. It is good to know the amount of improvement of the approach over a simple reordering model such as distance-based reordering (see Section 3.2.1), since even this simple model significantly outperforms a model without any reordering model.

[Collins et al., 2005] present a similar approach to [Xia and McCord, 2004], but with hand crafted rules to re-write the source sentence. They argue that baseline phrase-based models are unable to perform the reorderings such as those of between German and English. As they show, the main differences in German clause structure with English, it is clear that some of the reorderings require long distance skips which is usually penalised very high by phrase-based decoder, that makes it almost impossible to occur.

In addition, [Collins et al., 2005] highlight another benefit of source reordering which is able to bring together sets of words in the source sentence that can be extracted as a phrase, but without source reordering, they are not contiguous or they are far from each other to be considered as a phrase. They had six clause reconstructing rules which are sequentially applied to a German sentence. Sentences in the training set are reordered as well as test set. Phrases extracted by the baseline phrase extraction methods [Koehn et al., 2003] and test sentences are monotonically decoded. Following this work, [Wang et al., 2007a] apply the same method to Chinese, English pairs. Totally, eleven rules in three categories are selected and the gain of each rule in translation quality is examined. Despite a discussion in the paper about having a better word alignments with source reordering, it is still unclear how much source reordering helps to have better alignments.

A similar technique is used in [Badr et al., 2009] to reorder the source side in an English to Arabic statistical machine translation system. The difficulty of this work is that it is in the direction of generating a morphologically highly complex language such as Arabic. Their rules are mainly for Subject-Verb order in sentences that are translated into Verb-Subject in Arabic and noun phrase structures. Application of the rules require parsing the source side and syntactic reordering with poor quality parsers are not effective [Habash, 2007], however since the source language is English and there are reliable parsers for English, the reordering rules can be effective and the experiments show it.

Although all the above works reported improvements over the baseline, we believe reordering the source sentence makes hard decisions that eliminates the impact of n-gram language models. Thus, we prefer to make reordering decisions during the decoding.

In the above source reordering methods either a small set of rules were manually crafted [Collins et al., 2005] or a very large set of rules were automatically learnt [Xia and McCord, 2004]. An approach which automatically learns a small set of rules is presented in [Elming, 2008; Elming and Habash, 2009]. One important distinction between source reordering approaches is whether the source sentence is deterministically reordered and the decisions are made before the decoding or the reordered source sentence is used with the original sentence to be decoded. This non-deterministic method can also be done by producing a weighted lattice of source sentences, including the reordered ones and the original. [Elming, 2008] argue that a non-deterministic method is superior and propose a learning method to to learn reordering rules based on a set of linguistic information. In [Elming and Habash, 2009], a rule-based classifier is used to learn a small set of rules based on the linguistic features. A lattice is generated from all the possible reorderings permitted by the rule-set plus the original source sentence. The lattice is unweighted, however during the decoding the cost of the reorderings by the rules are estimated and taken into account for each hypothesis. This method of integrating the rule-based reorderings during the decoding enables the decoder to score the reorderings originated from the phrase-table too.

Another non-deterministic method, which uses an n-best list for providing the decoder with a list of reordered source sentences is proposed by [Li et al., 2007]. The train a maximum entropy model to decide whether two sister nodes in a binary syntax tree should be inverted or kept in the same order. Several features for each source phrase, including leftmost, rightmost, head, context words and their POS, were used in the maximum entropy binary classifier.

Another group of methods of source reordering are based on Part-of-Speech (POS) tags or statistically classified word classes[4]. [Chen et al., 2006] extract rules at the POS level from the word alignments and apply them to reorder the source sentences. [Rottmann and Vogel, 2007] use a combination of POS and POS with word context rules and a lattice as an input to the decoder. [Crego and Marino, 2006] extract rewrite patterns at POS level, however, instead of reordering the source sentence, the reordering operations are integrated into decoding process.

An approach that employs a method between full parsing and POS tags is [Zhang et al., 2007a] which is based on chunk-level. They apply a method similar to other source reordering methods, however in an intermediate level called syntactic chunks. A rule is composed of chunk and POS tags and word segmentation, POS tagging and chunking are the steps before extracting the rules. Table 3.1 shows some of the [Zhang et al., 2007a] rules which are extracted based on word alignments and source chunks of Chinese, English pair.

$NP_0\ PP_1\ u_2\ n_3$	0 1 2 3
$NP_0\ PP_1\ u_2\ n_3$	3 0 1 2
$DNP_0\ NP_1\ VP_2$	0 1 2
$DNP_0\ NP_1\ VP_2$	1 0 2
$DNP_0\ NP_1\ m_2$	0 1 2
$DNP_0\ NP_1\ m_2\ ad_3$	3 0 1 2
$DNP_0\ NP_1\ m_2\ ad_3\ v_4$	4 3 0 1 2

Table 3.1: Examples of reordering rules in chunk-based source reordering [Zhang et al., 2007a].

Due to the large number of the rules and also conflicting between some of them, a set of reordered sentences are passed to the decoder as a lattice. In their following work

[4] For a method to obtain bilingual word classes see [Och, 1999]

[Zhang et al., 2007b] the lattice weighting approach, which was based on only a trigram language model, is improved to incorporate the rules probabilities. In addition, they report further improvement by reordering the source sentences in the training data and extract an extra set of phrases to use along the original phrase table. In [Vilar et al., 2008] the extended the system to accept an n-best list instead of a lattice, which is claimed to have two advantages over the previous version. Firstly, it does not need a decoder which is able to process a lattice and a normal decoder can be used for the translation. Secondly, the number of reordering options in the n-best list is substantially lower than the lattice, but still it contains the best distinct paths, which increases the performance of the system.

3.2.3 Lexicalised Reordering Models

In lexicalised reordering models, a model is built to predict the word or phrase orientation during the decoding. These models assign a cost to the next candidate skip. The aim is to build a model that predicts the natural jump and penalise that jump less than other possible jumps by giving a lower cost to it. In this set of method, the models are mostly built based on word and phrase frequencies. A few simple syntactic features have been used in some of the proposed models, however the main source of evidence for these models are lexicalised statistics from the training data.

The two main branches of lexicalised reordering models are distortion models based on jumps between the words and phrase orientation models based on the orientation of the next phrase with respect to the last translated phrase.

Word-based Distortion Model

[Al-Onaizan and Papineni, 2006] argue that n-gram language models are not enough to deal with even local reorderings, thus they propose a distortion model to give a cost to each jump based on the words participated in the jump, The model computes the costs in word level, then combines the costs of the words to estimate the cost of the phrases. The model consists of three components: *outbound*, *inbound* and *pairwise* distortions.

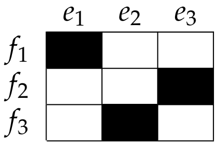
The outbound component tries to capture the skip length immediately after translating a particular word. Correspondingly, the inbound distortion captures the length of the jump before translating a particular word. The pairwise distortion considers both words to estimate the cost. Equations 3.2.2, 3.2.3 and 3.2.4 show the estimation of the probabilities of each distortion model for a jump with length δ between words f_i and f_j:

$$p_o(\delta|f_i) = \frac{\text{count}(\delta|f_i)}{\Sigma_k \, \text{count}(\delta = k|f_i)} \tag{3.2.2}$$

$$p_i(\delta|f_j) = \frac{\text{count}(\delta|f_j)}{\Sigma_k \, \text{count}(\delta = k|f_j)} \tag{3.2.3}$$

$$p_p(\delta|f_i, f_j) = \frac{\text{count}(\delta|f_i, f_j)}{\Sigma_k \, \text{count}(\delta = k|f_i, f_j)} \tag{3.2.4}$$

Here, count$(\delta|f_i)$ is the count of occurrence of a jump with length δ after word f_i. All the probabilities are directly estimated from word alignments. For example, in Figure 3.3, following counts will be incremented: $p_o(+1|f_1)$, $p_o(-1|f_3)$, $p_i(+1|f_3)$, $p_i(-1|f_2)$, $p_p(+1|f_1, f_3)$ and $p_p(-1|f_3, f_2)$.



	e_1	e_2	e_3
f_1	■		
f_2			■
f_3		■	

Figure 3.3: An example of frequencies which will be collected for word distortion model.

[Al-Onaizan and Papineni, 2006] convert the probabilities to distortion cost (log space), so they can be integrated into the phrase-based decoder as features. For example, pairwise cost is defined as:

$$C_p(\delta|f_i, f_j) = \log(\alpha P_p(\delta|f_i, f_j) + (1-\alpha)P_s(\delta)) \tag{3.2.5}$$

where $P_s(\delta)$ is a smoothing distribution[5] and α is set empirically. The experiments of

[5] In their experiments the distribution is a geometrically distribution.

[Al-Onaizan and Papineni, 2006] is on Arabic, English pair which they report improvements over the baseline. We implemented a distortion model based on their description and tested on German, English pair. Despite their report the improvements in our experiments were not significant over the distance based reordering model. In Chapter 4 there is a plan to find the reason for this.

Lexicalised Phrase Orientation Model

Inspired by the fact that some phrases are more likely to be reordered, lexicalised reordering models condition the reordering on the actual phrases. However, despite word distortion models, in phrase orientation models, instead of all possible reorderings between two particular phrases, only a set of reordering types are considered. The first lexicalised reordering model proposed by [Tillmann, 2004] which conditions the orientation of the phrase based on the phrase itself. There are three reordering types in their work: *Right*, *Left* and *Neither*. [Tillmann and Zhang, 2007] follow the previous work, by conditioning the orientation type on the previous and current phrase. Additionally, they add more features including a word distortion model based on the work by [Al-Onaizan and Papineni, 2006] (see Section 3.2.3). A stochastic gradient descent algorithm is provided to handle a large amount of features and predict the orientation of future phrases.

Similar to [Tillmann, 2004]'s work, [Koehn et al., 2005b] have three possible orientations: *monotone*, *swap* and *discontinuous*. The model is learnt directly for phrases from the alignment files. The probability $p_o(\alpha|\hat{f},\hat{e})$ is simply computed by counting how often a phrase pair is found with the particular orientation:

$$p_o(\alpha|\hat{f},\hat{e}) = \frac{count(\alpha,\hat{f},\hat{e})}{\Sigma_o count(o,\hat{f},\hat{e})} \tag{3.2.6}$$

where $\alpha \in \{monotone, swap, discontinuous\}$.

[Galley and Manning, 2008] argue that previous lexicalised reordering models fail capturing long distance reorderings and propose a hierarchical lexicalised reordering model. Despite dealing with hierarchical reordering rules, their method does not rely on cubic-time parsing algorithms such as those used in hierarchical phrase-based models (see [Chiang, 2005, 2007]). The model analyses the alignments beyond adjacent phrases to extract reordering rules, which are more complex than predicting the orientation between blocks of consecutive phrases. They classify lexicalised reordering models into word-based, phrase-based and hierarchical orientation models and demonstrate that the latter performs significantly better than the others.

[Tromble and Eisner, 2009] have considered reordering in machine translation as a case of Linear Ordering Problem and learnt the relative orders of words in a sentence based on multiple features. A dynamic programming algorithm based on chart parsing is developed to find the best reordering within a neighbourhood. They have used the method as a preprocessing step to translate German to English and reported improvements over a strong baseline equipped with a lexicalised reordering model.

[Zens and Ney, 2006] proposed a method based on maximum entropy principle to combine different features and predict the word orientation. They combined multiple lexicalised features and for generalisation, they considered features based on word classes. Although, they reported results for $\{-1, 0, 1\}$ possible jumps, their model is general enough to predict longer jumps. They concluded that features based on the source sentence words perform better than those based on the target and also more context always helps.

3.2.4 Hierarchical Phrase-based Model

Recalling the example in Section 2.3.2, one of the shortcomings of phrase-based models is disability to learn non-contiguous phrases. [Chiang, 2005] proposed a method to learn hierarchical phrases from the word alignments. In addition to learning non-contiguous phrases, this approach learns a set of synchronous grammar translation rules that can address the reordering problem in many cases. For example, in the German, English pair “Ich habe das Haus gekauft” and “I bought the house“, apart from normal phrases, we are able to extract “habe X gekauft”, “bought X” rule. In addition to the good phrase pair, it nicely captures the reordering of the X in the rule. Decoding process of synchronous grammar rules is different from the decoding process for phrase-based models (see Section 2.5). In phrase-based models we build the target sentence from left to right, however, here rules with gaps generate words in disconnected positions in the target sentence. Therefore, a chart parsing algorithm is used to decode the sentence by synchronous grammar rules. A full description of the decoding process is provided in [Chiang, 2007].

An extension to the string-based decoder is presented in [Galley and Manning, 2010] that allows discontinuous phrases such as those explained above in addition to continuous phrases be used without a CKY decoder. Their decoder [Cer et al., 2010] takes advantage of the better generalisations and reordering capabilities of the discontinuous phrases, which enables it to outperform both phrase-based decoders such Moses [Koehn et al., 2007] and hierarchical decoders such as Joshua [Li et al., 2009].

Summary

This chapter defined and explored the reordering phenomenon and the proposed approaches to deal with it in the literature. Local or short-distance and long-distance reorderings were discussed and it was argued that n-gram language models alone are sufficient to address the problem and several other models have been presented to compensate the lack of evidence provided by the language models. Many approaches and models have been proposed to deal with the problem. Syntax-based approaches rely on their syntactic rules to perform the reorderings and produce grammatically correct output. On the other hand, phrase-based approaches deal with most of the local reorderings with the help of extracted phrases and rely on additional features or preprocessing steps to tackle the rest of the reordering requirements.

We overviewed the lexical reordering models that are effective in phrase-based SMT decoders and also discussed the hierarchical versions of these lexicalised models. Also, some of the main syntax-based methods of SMT were presented that take a completely different approach to reordering and the the output fluency compared to the phrase-based models. We finished the chapter by the discussion of hierarchical phrase-based models and the integration of their translation model in the non-hierarchical phrase-based models.

CHAPTER 4

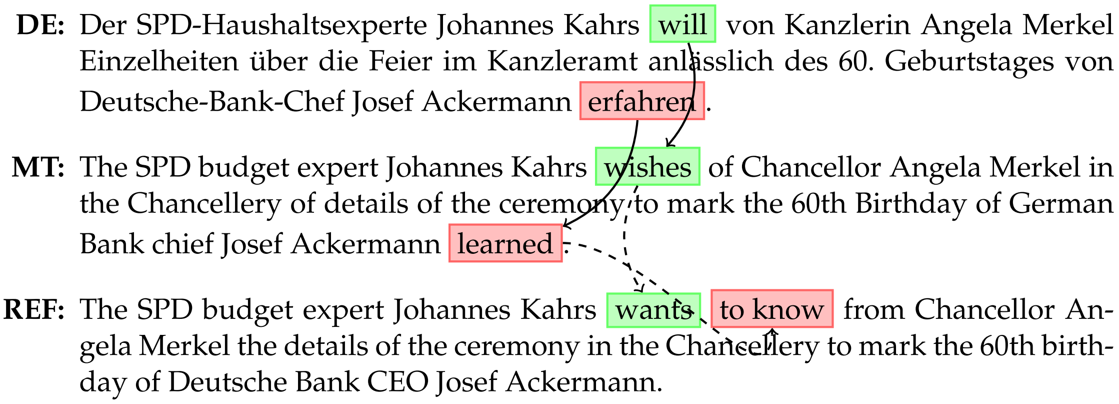
Decoding by Dynamic Chunking

4.1 Introduction

Despite the success of phrase-based statistical machine translation systems, fluency of the output, particularly for long sentences still remains one of the main challenges in current research on machine translation. Most of the errors in the MT output are caused by word-order differences between the source and the target language. In this chapter, we propose a method to guide the decoder in performing permutations and enable long distance reorderings required in many language pairs. The aim of the chapter is to outline an approach that is language independent and does not need any syntax-based language dependent tools. The method is called *dynamic chunking* and is motivated by the fact that words move together and groups of words can be translated without reorderings longer than those that can be captured by the phrase-table.

We have mentioned before that compared to word-based statistical machine translation systems, phrase-based approaches perform very well in capturing local reorderings. However, long distance reorderings remain a serious challenge. As Knight [1999] showed, trying all the permutations is computationally intractable, and most phrase-based MT systems restrict the search space by limiting the set of reorderings that are explored during decoding. Zens et al. [2004] examine the effect of different constraints on machine translation quality.

A constraint commonly used in phrase-based machine translation is the distortion limit, which restricts the distance between the next phrase and the previously translated phrase. Most approaches described in the literature report a distortion limit ranging between 4 and 12 words. This limitation of course prohibits any word reordering going beyond the set limit. This might not be a problem for language pairs with similar word order such as English-French or Dutch-German [Birch et al., 2008]. A good language model or a lexicalised reordering model [Koehn et al., 2005a] will be enough to capture the word order differences in these cases. However, when translating between languages with rather different word order, for example an SOV (subject-object-verb) language into an SVO (subject-verb-object) language, the distortion limit restriction can severely affect the decoder's ability to capture those word order differences correctly. When translating from German (an SOV language) into English (an SVO language), it is not unusual that more than 20 words on the source side need to be jumped over to translate the verb in the right position. Figure 4.1 shows a German sentence translated into English. The SMT decoder can not easily skip the distance between `will` and `erfahren` to correctly translate them into `wants to know`. The two German phrases are likely to separately be translated and hence generate a non-fluent English.



DE: Der SPD-Haushaltsexperte Johannes Kahrs will von Kanzlerin Angela Merkel Einzelheiten über die Feier im Kanzleramt anlässlich des 60. Geburtstages von Deutsche-Bank-Chef Josef Ackermann erfahren .

MT: The SPD budget expert Johannes Kahrs wishes of Chancellor Angela Merkel in the Chancellery of details of the ceremony to mark the 60th Birthday of German Bank chief Josef Ackermann learned .

REF: The SPD budget expert Johannes Kahrs wants to know from Chancellor Angela Merkel the details of the ceremony in the Chancellery to mark the 60th birthday of Deutsche Bank CEO Josef Ackermann.

Figure 4.1: A German sentence that requires a long distance reordering to correctly translate the verb. DE is the German sentence, MT is the output of the machine translation system and REF is the human translation.

While relaxing the distortion limit accordingly may seem a possible solution to this problem, it has two severe shortcomings: Firstly, decoding time rapidly increases with more relaxed distortion limits. Secondly, wider distortion limits also allow for *any* reordering within the distortion limit which increases the level of noise and puts a higher burden on the language model to demote wrong reorderings.

In this chapter, we propose a method to enable the decoder to consider permutations which include long distance reorderings. By grouping words and moving them together, we try to enable the decoder to consider long-distance reorderings and avoid unnecessary short distance permutations. In addition, our method does not rely on language-dependent parsers or chunkers and uses the word alignment information to build the chunker. In this chapter we use the term chunk for contiguous group of words. In phrase-based SMT models, a phrase is also a span of words, however there are several differences between a phrase and a chunk. Firstly, the purpose of chunking a sentence is to find a group of words that can be translated monotonically, but phrases are extracted from the word alignment data regardless of the word orders. Secondly, chunks may contain several phrases and therefore they are designed to be longer than phrases, so multiple phrases can be translated during a chunk translation. Thirdly, the method of identifying chunks, presented in Section 4.3.5 is different than the phrase extraction algorithm. Finally, the chunks are only used to guide the decoder in reordering decisions and are not used for word replacements. On the other hand, the main use phrases is to replace the source sentence with target words.

The rest of the chapter is organised as follows: Section 4.2 provides an overview of the related work addressing the issue of word reordering in statistical machine translation and the use of chunking in particular. Section 4.3 explains the proposed method. Section 4.4 discusses the experimental settings and results comparing the chunking method to a baseline. In Section 4.5 we draw some conclusions and discuss open issues. Furthermore, Section 4.5 analyses the shortcomings of the approach proposed in this chapter and suggests a few extensions and modifications to improve the quality of this approach.

4.2 Related Work

We explained in the previous chapter that several phrase-based SMT systems use a very simple distance-based reordering model [Koehn et al., 2003, 2007]. In such a distance-based model, monotone translation and short jumps are preferred over longer jumps. The cost in this model increases linearly by distance with a slight preference for jumps to the right:

$$d(i) = start_i - end_{i-1} - 1 \tag{4.2.1}$$

where $d(i)$ is the distortion cost of translating the ith phrase after the $(i-1)$th.

More recently, there have been efforts to incorporate syntax into statistical machine translation, or using syntactic means in order to address the issue of word reordering. A method to incorporate syntactic information is to apply syntactically motivated rules to render the word order of the source sentence similar to the target language. These transformation rules can be syntax-based or lexicalised rules. A syntax-based rule is a transformation rule that only contains syntactic tags [Collins et al., 2005; Wang et al., 2007a], but a lexicalised rule contains at least one word as a constraint [Xia and McCord, 2004]. Xia and McCord [2004] proposed a method to learn transformation rules, lexicalised and syntax-based (unlexicalised), from a parallel corpus. Their approach extracts re-write patterns, applies them to the source sentence after which the sentence is translated monotonically. To learn the rewrite patterns, the source side of the bi-text is parsed, phrases are aligned and lexicalised, and unlexicalised patterns consisting of parent nodes with their children, plus their syntactic labels are extracted.

Zhang et al. [2007a] developed a method similar to other source reordering methods, however their approach works on an intermediate level called 'syntactic chunks'. A syntactic chunk is a series of words that consist of a grammatical unit such as noun and verb. They use a maximum entropy tool to build the chunking model with training data provided by converting sub-trees of Chinese treebank into chunks. A rule is composed of chunk and POS tags, where a chunk tag for each word determines the chunk type that the word belongs to and also whether the word is at the beginning of the chunk. Before extracting the rules POS tagging and chunking is applied. As several conflicting rules can match a given sentence, the different rule applications are passed to the decoder as a lattice.

For all of the the reordering approaches discussed above, a syntactic parser, chunker, or POS tagger of the foreign language is required. Unfortunately, these resources (at sufficient levels of accuracy) tend to be scarce for many languages.

On the other hand, we believe reordering the source sentence makes hard decisions that cannot be undone. For example, Xia and McCord [2004] report a decrease in translation quality by allowing permutations after reordering the source sentence. Also, since all reorderings are done beforehand, the impact of n-gram language models, which is quite crucial in other approaches, is eliminated. To take advantage of the language model feature, we prefer to make reordering decisions during decoding. In addition, since one of the strengths of phrase-based models is to learn many phrases which do not necessarily belong to any syntactic category [DeNeefe et al., 2007], we believe the syntactic chunks may diminish this feature. Therefore, we suggest to consider all possible chunks and identify the optimal chunk boundaries during decoding.

There are also a number of reordering approaches that fully integrate reordering into the decoding process, see for example [Al-Onaizan and Papineni, 2006; Tillmann, 2004] as mentioned in Chapter 3. These models typically predict the jump orientation (and sometimes distance) based on the previously translated phrase and the phrase that is to be translated next. A few simple syntactic features have been used in some of these models [Crego and Marino, 2006], however the fully lexicalised parameters remain the main source of evidence. Our method differs from lexicalised reordering models as it allows permutations beyond the fixed distortion limit and also removes the need for considering many unnecessary local reorderings.

4.3 Integrating Chunking and Decoding

In this section we describe our approach which integrates chunking and decoding. While all of the previous chunk-based decoders first apply chunking, then reorder the chunks, and finally perform translation, our approach performs chunking and decoding at the same time. The advantage is that decisions at each level (chunking, chunk-based reordering, and translation) are not made independently of each other.

Penalising the jumps according to the number of words in distance-based reordering severely discourages making long distance reorderings and tends to bias the decoder to translate most of the sentences monotonically [Collins et al., 2005]. Here, we group words together and penalise the jumps based on the number of skipped chunks. This enables the decoder to skip more than a fixed number of words and allows for long-distance reorderings. On the other hand, we chunk the source sentence in a way that words inside a chunk can be translated monotonically in either direction: right to left or left to right. By eliminating local reorderings (apart from the local reorderings that are captured by the phrase translations themselves) within the chunks the size of the search size is kept manageable during decoding.

To accomplish this, we extended the standard phrase-based multi-stack decoding approach to simultaneously chunk and apply phrase applications. The approach consists of two components: Firstly, a chunk scoring component which is a binary classifier that gives each chunking candidate a score, and, secondly, an extension to the decoder that either expands the current chunking decision or applies a phrase translation inside an uncovered chunk.

We use a *maximum entropy* classifier to assign score to each chunking decision. In the section, we first briefly discuss the principle of maximum entropy and describe how the features are defined to be used in a classifier based on maximum entropy and then explain the chunking scorer in detail.

4.3.1 Maximum Entropy Modeling

Statistical modelling is used to build a model to predict the behaviour of a process. A labelled training set is employed to learn a model predict future behaviour of the process [Berger et al., 1996]. The first modelling task is feature selection and the second one is model selection. Firstly, a set of statistics is determined and then these statistics will be employed to construct an accurate model of the desired process.

One of the approaches to build that model is through maximum entropy modelling. The idea behind maximum entropy method is very simple: model all that is known and assume nothing about that which is unknown [Berger et al., 1996]. It means, choose a model consistent with all the facts, but otherwise as uniform as possible.

Calculating the model for a very limited number of constraints is easy and the problem can be solved by simple mathematical operations. However, when there are many constraints—which is typically the case in NLP problems—it can not be solved analytically and numerical methods have to be leveraged to find the model.

4.3.2 Representing Evidence

Consider a random process that produces output values $y \in \mathcal{Y}$ which may be influenced by contextual information $x \in \mathcal{X}$. The model that represents this process is a method to estimate the conditional probability $p(y|x)$, where $p(y|x)$ is the probability of seeing output y in presence of context x. On the other hand, take $\tilde{p}(y|x)$ as the empirical distribution of some number of samples $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ which are observed from the behaviour of the process. These samples, which are *constraints* on distribution p, are extracted from training data. Even though large training sets usually contain some occurrences of y and x together they are usually not sufficient to predict $p(y|x)$ for any (x, y) pair. Thus, one side of the problem is finding a distribution which matches the constraints. To express these facts, any statistic from the samples is introduced as a function $f(x, y)$ which is called a *feature*:

$$f_i(x,y) = \begin{cases} 1 & \text{if} \quad c(x,y) \\ 0 & \text{otherwise} \end{cases} \tag{4.3.1}$$

where f_i is a function which is 1 if the predicate c is *true* for some x and y.

Although in some natural language processing tasks features are binary functions of the form given in Equation (4.3.1), in many classification tasks, including text classification, usually the strength of evidence is taken into account. In other words, to represent features, instead of a binary function that indicates the presence or absence of a contextual information, a real-valued function is used to indicate the strength of context. Nigam et al. [Nigam et al., 1999] have shown that using strength of evidence in text classification increases performance.

4.3.3 Maximum Entropy Principle

The probability distribution for the process based on maximum entropy has two characteristics: Firstly, it is in accordance with the constraints, secondly it is as uniform as possible. The first statement means the model's expected value of the feature f_i is equal to the observed expectation of that feature. Thus, if $E_p x$ is p's expected value of x and we have n features $\{f_1, f_2, ..., f_n\}$, then:

$$E_p f_i = E_{\tilde{p}} f_i \quad \text{for} \quad i \in \{1, 2, ..., n\} \tag{4.3.2}$$

Equation (4.3.2) is called a *constraint*. $\tilde{p}(y|x)$ is the observed probability of y given x in the training data. $E_p f_i$ is the expected value of f_i in model p, which is equal to $\sum_{x,y} \tilde{p}(x) p(y|x) f_i(x,y)$, where $\tilde{p}(x)$ is the empirical distribution of x in the training sample. For the second statement, suppose $\mathcal{P}$ is the set of all possible probability distributions and $\mathcal{C}$ is the subset of $\mathcal{P}$ which are compatible with constraints:

$$\mathcal{C} = \{p \in \mathcal{P} | E_p f_i = E_{\tilde{p}} f_i \quad \text{for} \quad i \in \{1, 2, ..., n\}\} \tag{4.3.3}$$

According to the maximum entropy principle we have to select the most uniform probability distribution in $\mathcal{C}$. The measure of uniformity for the conditional probability $p(y|x)$ is [Berger et al., 1996]:

$$H(p) \equiv -\sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \tag{4.3.4}$$

So, the required distribution is the maximum of $H(p)$:

$$p^* = \underset{p \in \mathcal{C}}{\arg\max}\, H(p) \tag{4.3.5}$$

It can be shown that there is a unique distribution that satisfies Equation 4.3.5 and it is always of the exponential form. Berger et al. [Berger et al., 1996] have shown that the solution has the following parametric form:

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} \exp \sum_{i=1}^{n} \lambda_i f_i(x,y) \tag{4.3.6}$$

$$Z_\lambda(x) = \sum_y \exp \sum_{i=1}^{n} \lambda_i f_i(x,y) \tag{4.3.7}$$

where $Z_\lambda(x)$ is a constraint to satisfy the requirement that $\sum_y p_\lambda(y|x) = 1$ for all x, because it is a probability distribution.

4.3.4 Optimisation Algorithm

Apart from simple problems, Equation 4.3.6 can not be solved analytically and numerical methods should be used to find the weights of the features. There are many algorithms for estimating parameters of maximum entropy models. Generalised Iterative Scaling [Darroch and Ratcliff, 1972] and its improved version Improved Iterative Scaling [Berger et al., 1996] are widely used to optimise maximum entropy models, however, in our experiment both algorithms performed very poorly which is also reported by Malouf [Malouf, 2002]. Thus we decided to use Nocedal's limited-memory BFGS [Nocedal, 1980] which is a very efficient and robust method to solve large scale optimisation problems [Liu and Nocedal, 1989]. L-BFGS is a version of quasi-Newton method BFGS, which is provided to overcome the memory problem of BFGS algorithm [Nocedal, 1980].

The L-BFGS algorithm has been implemented in Andrew McCallum's MALLET library [McCallum, 2002] and Zhang Le's Maximum Entropy Toolkit [Le, 2004].

4.3.5 Chunking Scorer

We define a chunk as a contiguous group of words that can be translated monotonically from left to right or right to left. Figure 4.2 shows an alignment matrix for a pair of sentences. Given a word alignment a_1^J between a source sentence $\mathbf{f} = f_1, ..., f_J$ and target sentence $\mathbf{e} = e_1, ..., e_I$. We define a *chunk boundary* between f_j and f_{j+1} if there is a source word aligned to any i such that $a_j < i < a_{j+1}$ or $a_j > i > a_{j+1}$. Formally:

Definition 1. *Suppose A_j is the set of all a_js, which is the set of all the positions that are aligned to f_j. We define a chunking boundary between f_j and f_{j+1} if there is a source word aligned to any i such that $max(a_j, a_{j+1}) > i > min(a_j, a_{j+1})$, where $a_j \in A_j, a_{j+1} \in A_{j+1}$ and $|a_j - a_{j+1}|$ is the minimum*[1].

For instance, in the example alignment, there is no *chunk boundary* between f_6 and f_7, because there is no i such as $a_6 < i < a_7$. Analogously for f_1 and f_2, as there is no source word aligned to e_2. According to this definition there is, for example, a *chunk boundary* between f_2 and f_3. The example in Figure 4.2 contains three chunks. With this definition, a binary classifier will be learnt to classify every point between two foreign words under two classes: 'chunk boundary' and 'no chunk boundary'

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
e_1	■						
e_2							
e_3		■			■		
e_4				■			
e_5			■				
e_6						■	
e_7							■

Figure 4.2: An example of chunks with left to right, (f_1, f_2), (f_6, f_7) and right to left (f_3, f_4, f_5) orientations.

We define a set of features based on the word alignments and above definition to be used in the maximum entropy classifier. Our set of feature functions include:

- $h_1(\delta, f_j, f_{j+1})$, where $\delta \in \{1, 0\}$, $+$ indicates that the words are in different chunks, so the point between them is a chunk boundary. h_1 gives the probability of being a chunk boundary or not based on the collected frequencies. In the example of

[1] The following conditions ensure that the definition is consistent for positions with source words that are aligned to more than one target words, which is A_j with more than one element: $a_j \in A_j, a_{j+1} \in A_{j+1}$ and $|a_j - a_{j+1}|$ is the minimum

Figure 4.2, we increment the count$(1|f_2, f_3)$, count$(1|f_5, f_6)$ and for all the other pairs count$(0|f_j, f_{j+1})$.

- $h_2(\delta, f_j)$, where $\delta \in \{1, 0\}$, 1 indicates the word is a left border of a chunk. In the example, f_1, f_3 and f_6.
- $h_3(\delta, f_j)$, where $\delta \in \{1, 0\}$, 1 indicates the word is a right border of a chunk. In the example, f_2, f_5 and f_7.
- $h_4(f_j, f_{j+1})$, which is a binary function indicating the significance of the pair in the data.

Given the above feature functions, a first set of training sentences is used to collect the lexicalised frequencies and train the model, the second part is used to generate features for parameter estimation of the maximum entropy classifier. We use L-BFGS [Nocedal, 1980] implemented in [Le, 2004] to optimise the feature weights.

The above feature functions are combined to estimate the chunking scores in Equation 4.3.6 and the chunking scorer is integrated into the baseline decoder as an additional feature. The feature function to integrate into the decoder is:

$$\begin{aligned} h_{chunk}(f_1^J, e_1^I, C, S) = \\ \log \prod_1^J (C_j S(j) + (1 - C_j)(1 - S(j))) \end{aligned} \qquad (4.3.8)$$

where C is a function that maps each position on the foreign side to the set $\{1, 0\}$, indicating whether there is a chunk boundary after this word. S is the chunking scorer that assigns to each position the probability of being a chunk boundary.

4.3.6 Decoding by Chunking

The decoder is a multi-stack, multi-beam decoder that translates the sentence from left to right, which can skip multiple chunks and translate them later to perform any kind of reordering. For expanding each hypothesis either an uncovered chunk is picked and a phrase translation is applied or a new location is marked as a chunk boundary. As the chunking decisions affect the way phrase translations are applied, we insert hypotheses with the same covered words and the same last chunked position in the same stack. For expanding each hypothesis, the first step is to label more chunks from the last chunked position, which means expanding the current hypothesis by finding more chunks and assigning to them the chunking cost. In the next step, if the current position is inside an uncovered chunk, the decoder continues translating the chunk by applying new phrase translations. Otherwise, it picks a new chunk to translate and starts applying phrase translations within the chunk. No reordering inside the chunks is allowed. Figure 4.1 sketches the algorithm.

```
Input: $multiStack$
for all $hyp$ in $multiStack$ do
  $current$ ← find current chunk {based on last foreign position}
  if $current$ is completely covered[2] then
    $chunksToExpand$ ← chunk limit − available chunks
    for $i = 0$ to $chunksToExpand$ do
      create new hypothesis from $hyp$ and $i$
    end for
    for all $chunk$ in uncovered chunks do
      start translating $chunk$
    end for
  else
    continue translating $current$
  end if
end for
```

Algorithm 4.1: Decoding and chunking integration algorithm.

Figures 4.3 and 4.4 show an example of a chunk based derivation. Figure 4.3 is the step by step derivation of the target sentence and Figure 4.4 shows the chunk movements from the source sentence to the target sentence.

In state 1 of this example, the decoder labels the position between German words ‘muss’ and ‘die’ as a chunk boundary. This is a chunking state (C), which finds the labels of the positions between the words and computes the chunking cost by the chunking scorer component. For the next state, the decoder either labels more positions to be chunked or applies phrase translations to uncovered words. The latter is

[2] all the words in the chunk are translated

done by translating the span ‘man muss’. A translation state (P) can be reached by multiple phrase applications. In states 3 and 4, more positions are labelled as chunk boundaries (between ‘wirkung’, ‘anerkennen’ and ‘anerkennen’, ‘.’). In the next state, the decoder jumps over a chunk (9 words) to translate the verb. Grouping the words together makes it possible to do long-distance reordering such as this. The remainder of the decoding process is to translate the skipped chunk monotonically and finally chunk and translate the full stop.

With extra information in every hypothesis, the recombination criteria are redefined to consider the chunking status of a hypothesis. For two hypotheses to be recombinable [Koehn, 2004], they should have identical chunk boundaries for the uncovered positions. This is in addition to commonly used recombination criteria such as identical cover vectors, language model history, and last foreign position covered.

The chunking cost, estimated by the chunking scorer, is another feature along the baseline features. Also, the future cost computation component includes the future chunk distortion cost and future chunking cost together with the translation model and language model costs.

The following feature functions are defined to incorporate chunking costs and chunk reorderings costs:

- Chunking cost feature function which assigns to each chunk a probability according to the classifier explained in the previous section.
- Chunking penalty which penalises or rewards each chunking application based on the sign of its weight. The optimisation algorithm, configures this feature in a way to encourage or discourage longer chunks.
- Chunk distortion model which penalises jumps over chunks similar to distance-based reordering model, however instead of the number of words, it counts the number of chunks.

1	[man muss] die schwierigkeiten bei der bestimmung von ursache und wirkung anerkennen .
C	
2	[**man muss**] die schwierigkeiten bei der bestimmung von ursache und wirkung anerkennen .
P	we must
3	[man muss][die schwierigkeiten bei der bestimmung von ursache und wirkung] anerkennen .
C	we must
4	[man muss][die schwierigkeiten bei der bestimmung von ursache und wirkung][anerkennen] .
C	we must
5	[man muss][die schwierigkeiten bei der bestimmung von ursache und wirkung][**anerkennen**] .
P	we must recognise
6	[man muss][**die schwierigkeiten bei der bestimmung von ursache und wirkung**][anerkennen] .
P	we must recognise the difficulties in the provision of cause and effect

Figure 4.3: An example of the decoding process by dynamic chunking. The *C* states are chunking states, which new chunking boundaries are detected and in *P* states, phrase translations are applied inside a chunk. The bold parts of the source sentence show the translated spans in that state. The rest of the decoding is chunking and translation the full stop.

- **DE** man muss die schwierigkeiten bei der bestimmung von ursache und wirkung anerkennen .
- **REF** the difficulties in determining cause and effect must be acknowledged .
- **BL** it must be the difficulties in the provision of cause and effect .

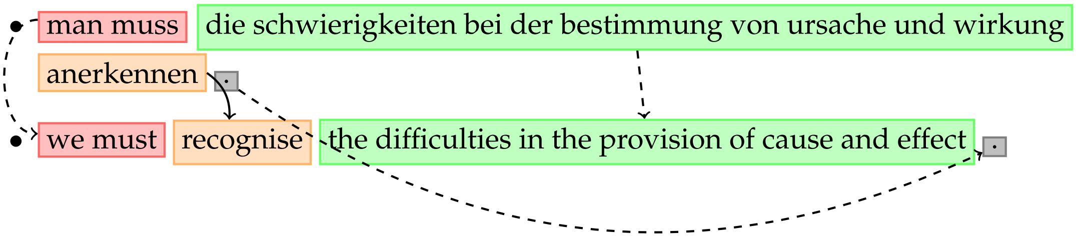

man muss
die schwierigkeiten bei der bestimmung von ursache und wirkung
anerkennen
.
we must
recognise
the difficulties in the provision of cause and effect
.

Figure 4.4: An example of chunk movements during the decoding by dynamic chunking. DE is the German sentence, REF is the reference translation and BL is the translation by the baseline decoder.

4.3.7 Parameters

To control the quality and the speed of the decoder for different language pairs, a few additional parameters are introduced. Since decoding inside the chunks is monotone, all baseline parameters[3] apart from the distortion limit are also needed here.

- chunk length limit: determines the maximum allowed length for each chunk. A large value, such as 100, lets the decoder try all available chunks. On the other hand, for languages with many local word reorderings a smaller value can make the decoding process faster without hurting the performance (Default: 100).
- chunk number minimum and maximum: These values control the number of uncovered chunks before applying phrase applications. They can be used to control the amount of permutations during decoding (Default: 1 and unlimited).
- chunk distortion limit: similar to distortion limit in the baseline, but based on the chunks instead of words (Default: 6).

[3] This includes: stack limit, beam width, phrase length limit, and phrase table entries per source phrase.

		German	English
Train	Sentences	1.4M	
	Words	38M	40M
	Vocabulary	344K	113K
	Avg Sen. Length	26.17	27.51
Test(EP)	Sentences	2,000	
	Words	56K	60K
	Vocabulary	8844	6050
	Avg Sen. Length	28.31	30.09
Test(NC)	Sentences	2,028	
	Words	51K	49K
	Vocabulary	9849	7163
	Avg Sen. Length	25.31	24.63

Table 4.1: German to English corpus statistics. Europarl (EP) and News Commentary (NC) test sets of ACL WMT 2008.

4.4 Experiments

4.4.1 Experimental Setup

To examine the effects of dynamic chunking on translation quality, we have chosen German to English translation as it involves many long distance reorderings. The training and test data sets are taken from the ACL WMT evaluation [Koehn and Monz, 2006]. The corpus statistics are shown in Table 4.1.

The preprocessing stage includes tokenisation and lower casing. The tokenisation process separates words. In English and German words are mostly separated from each other by whitespace. Our tokenisation algorithm uses whitespace and punctuations and some simple rules for exceptions to tokenise the text. There is only one reference translation for each sentence. The evaluation metrics used here are BLEU [Papineni et al., 2001], NIST [Doddington, 2002] and TER [Snover et al., 2006].

The baseline system is a common multi-beam, multi-stack phrase-based decoder, described in [Koehn et al., 2003] with the following features:

- phrase translation probabilities and lexical probabilities for both directions

- a trigram language model

	Run	System	BLEU	NIST	1−TER
1	EP	Baseline	0.2687	7.0063	0.3374
2	EP	Chunk	0.2716	7.1084	0.3261
3	NC	Baseline	0.2454	7.1591	0.3476
4	NC	Chunk	0.2487	7.1798	0.3599

Table 4.2: Results on German to English task of ACL WMT 2008 translation task, Europarl (EP) and News Commentary (NC) test sets. Since TER is measuring the error, 1−TER is reported. Default values are used for parameters of the chunking decoder (see Section 4.3.7).

- phrase and word penalty
- distance-based reordering penalty

The weights for the features are optimised by MER training [Och, 2003] to maximise the BLEU [Papineni et al., 2001] score.

4.4.2 Results

The maximum entropy classifier is evaluated on the held-out data of the parallel corpus. The average accuracy[4] of 10-fold cross validation is 0.73, which means that around 25% of the chunk boundary decisions are incorrect. On the other hand, the classification decisions are not the only source of evidence that we use to choose the chunking boundaries. Both the language model and the translation models (phrases that cover the span) contribute to this decision. The probability of being a chunk boundary in the training data is 0.3, which is nearly identical to the probability of assigning a chunk boundary during the decoding. However, in 32% of the cases the chunking decision during decoding differs from the decision of the maximum entropy classifier. This means, even though the classifier classifies a point as a chunking boundary, the decoder decides not to use that chunking decision, mainly based on the translation and language model costs.

Table 4.2, shows the results of the chunking approach compared to the baseline. By looking at the translation outputs of the chunking system and comparing it to the base-

[4] The accuracy is computed based on how many of the boundary points are classified correctly. Note that, a sentence of length J, has $J-1$ boundary point.

- **DE** die anstrengungen können nicht von den erzeugern allein unternommen werden .
- **REF** efforts cannot be made by producers alone .
- **BL** the efforts made by producers alone cannot be done .

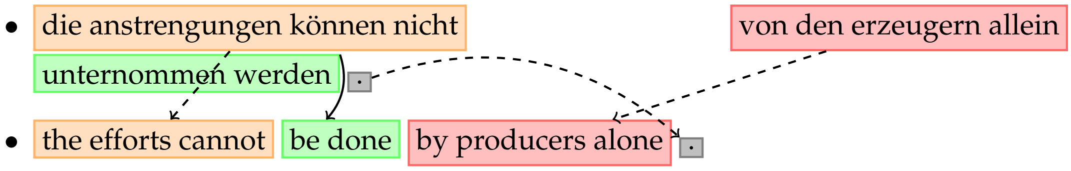


Figure 4.5: An example of useful chunk movements by the dynamic chunking decoder. For the order of translation refer to Figure 4.3. DE is the German sentence, REF is the reference translation and BL is the translation by the baseline decoder.

line, we can observe that the chunking system generates very different translations to the baseline and not in all cases captures the proper order of the chunks to translate. In general, there are three main reasons for the chunking system to fail. Firstly, a wrong classification decision by the chunking scorer may lead the decoder to jump or monotonically translate in a wrong position. Secondly, although the classifier picks a proper chunking boundary, the other features force the decoder to apply the wrong reordering. Finally, even with accurate chunk boundaries, the decoder can still fail to apply the correct reorderings.

4.5 Discussion and Error Analysis

Inspired by previous work on integrating syntactic chunking into machine translation, a decoder that dynamically chunks and translates the source sentences is developed. The results show that the chunking system generates very different translations compared to the baseline and it is effective for a language pair such as German to English that needs long-distance reorderings. Dealing with data sparseness and more accurate classification for detecting chunking boundaries seems very promising.

Although the current set of classification features is quite simple and it does not contain word classes or POS features, it performs well compared to the baseline. Incorporating more features and using word classes to deal with data sparseness could result in better classifier decisions and higher translation quality. It is not entirely surprising that the language model seems insufficient to accurately distinguish between correct and incorrect reorderings of chunks in all cases. A lexicalised reordering model on the chunk-level could help to improve this aspect of our approach.

Figures 4.3 shows examples of dynamic chunking derivations compared to the baseline derivation and their reference translations. Figure 4.5 shows the chunk movements in for the last example in Figure 4.3. After error analysis, we categorised the shortcomings of the dynamic chunking approach into three classes: bad chunking decisions, confusion in chunks translation order and slower decoding compared to the baseline and difficulties in optimisation. We discuss each issue separately and propose our solutions to each one of them.

- **Bad chunking decisions**: as mentioned in Section 4.4.2 the classifier has 0.73 precision in finding the ideal chunking boundaries. The ideal chunking boundaries are based on the word alignment data and are not necessarily the best options, nevertheless, in 25% of the times the classifier does not find the optimal solution based on the learned model[5]. On the other hand, in some cases the ideal chunking boundary based proposed by the classifier is not a good boundary. There are many cases that the decoder chooses a chunking boundary between two words that can be covered by a useful phrase. One main reason for this problem is the chunking boundary definition provided in Section 4.3.5. This definition is based on the word alignments data and relies on the word statistics. However, the decoding process only deals with the phrases and does the reordering based on the phrases.

  To modify the definition and the classifier, we need a set of training data that contains phrase alignment. To prepare this data, we modify the decoder to find translations motivated by the reference sentences. Basically, we want to find the closest translation that is possible to generate by the phrase-table and the decod-

[5] It is called *search error*.

ing algorithm. To score each hypothesis, we consider the smoothed BLEU score[6] contributed by this hypothesis as its score. Smoothed BLEU is a modification of BLEU measure (see Section 2.6 in chapter 1), so it can be used on one sentence basis. This method provides us the phrase alignment that we need to define the chunking boundaries and also train the chunking classifier.

In addition to the revising the definition, another feature can be integrated into the chunking scorer. Syntactic chunks alone are not very useful in capturing the reordering requirements of machine translation, however, integrating them with the current classifier might incorporate useful syntactic information into the reordering decisions.

- **The order of translating the chunks**: the chunk expansion algorithm described in Section 4.3.6, relies on the language model to find the order of translating the chunks. To address this problem, we propose to build a model based on the first word of each chunk or its POS tag. Therefore, a lexicalised model that scores *monotone*, *swap* or *discontinuous* of chunks.

- **Slower decoding and difficulty in optimisation**: the main issue here is redundant hypotheses which are considered by the decoder and will be recombined in future. Basically, because the dynamic chunking decoding adds another dimension to the multi-stack, there are more similar hypotheses with different derivations. This not only hurts the performance of the decoding, it also fills the *n*-best list, used for optimisation, with similar translations and different derivations. One approach to deal with this problem is having a graph of possible chunking decisions (for the entire sentence) and use a 2 dimensional stack like the baseline's. Based on graph, the path of decoding is determined and redundant derivations are avoided.

[6] Any other evaluation measures can be used.

Summary

In this chapter we presented an extension of a phrase-based decoder that dynamically chunks, reorders, and applies phrase translations in tandem. A maximum entropy classifier is trained based on the word alignments to find the best positions to chunk the source sentence. No language specific or syntactic information is used to build the chunking classifier. Words inside the chunks are moved together to enable the decoder to make long-distance reorderings to capture the word order differences between languages with different sentence structures. To keep the search space manageable, phrases inside the chunks are monotonically translated, thus by eliminating the unnecessary local reorderings, it is possible to perform long-distance reorderings beyond the common fixed distortion limit. Experiments on German to English translation are reported.

CC	[sie kann nicht][als grundlage für die einführung einer europäischen verfassung][dienen][.]
CD	[1 sie kann nicht 1][3 als grundlage für die einführung einer europäischen verfassung 3] [2 dienen 2][4 . 4]
RE	it cannot serve as a basis for the establishment of a european constitution .
BL	it is not as a basis for the introduction of a european constitution .
CH	it cannot serve as a basis for the introduction of a european constitution .
CC	[ich weiß , dass es] [bezüglich des einen oder anderen änderungsantrags noch meinungsverschiedenheiten gibt][.]
CD	[1 ich weiß , dass es 1][4 bezüglich des einen oder anderen änderungsantrags 4] [3 noch meinungsverschiedenheiten 3][2 gibt 2][5 . 5]
RE	i know there are still differences of opinion on this or that amendment .
BL	i know that it is on the one or other amendment still differences of opinion .
CH	i know that there are still differences of opinion with regard to the one or other of the amendment .
CC	[ich möchte][frau gebhardt zu einer guten arbeit][gratulieren][.]
CD	[1 ich möchte 1][3 frau gebhardt zu einer guten arbeit 3][2 gratulieren 2][4 . 4]
RE	i would congratulate mrs gebhardt on a good piece of work .
BL	i would say to mrs gebhardt on a job well done .
CH	i would like to congratulate mrs gebhardt on a job well done .
CC	[hoffen wir][, dass wir][künftig diese garantien][erreichen können]
CD	[1 hoffen wir , dass wir 1][3 künftig diese garantien 3][2 erreichen können 2]
RE	let us hope that in the future we will be able to achieve those guarantees
BL	let us hope that in future we can achieve those guarantees
CH	let us hope that we can achieve these guarantees
CC	[die anstrengungen können nicht][von den erzeugern][allein][unternommen werden][.]
CD	[1 die anstrengungen können nicht 1][3 von den erzeugern allein 3][2 unternommen werden 2][4 . 4]
RE	efforts cannot be made by producers alone .
BL	the efforts made by producers alone cannot be done .
CH	the efforts cannot be done by producers alone .

Table 4.3: A few translation samples comparing the chunking-based decoder and the baseline. CC indicates the chunking decisions by the maximum entropy classifier. CD are the chunking boundaries picked by the decoder and their order of translation. RE is the English reference sentence. BL is the baseline output and CH is the chunking-based decoder output.

CHAPTER 5

Dynamic Distortion in a Discriminative Reordering Model

In the dynamic chunking approach explained in Chapter 4, the focus was on the long distance reordering capabilities of the decoder and the short distance permutations were left to be handled by the phrase-table. That approach is effective for translation from languages such as German to English that need long distance reorderings because of structural differences in their word order[1]. Among the shortcomings of the dynamic chunking approach was the lack of local reordering in positions that are not captured by the phrase-table.

In this chapter, we equip the decoder with a discriminative reordering modelling that combines several features representing the context of the hypothesis expansion. The reordering model helps the decoder to make better reordering decisions in general. In addition, we improve the decoding by dynamically adjusting the reordering window, so without ignoring the local permutations it becomes possible to make long distance jumps.

As mentioned in Chapter 3, an important parameter in most phrase-based systems, which controls the size of the search space explored by the decoder, is the so-called *distortion limit*. The distortion limit specifies the size of the window which the decoder considers to choose the next source phrase. The best value for this parameter is dif-

[1] Subject-Object-Verb to Subject-Verb-Object

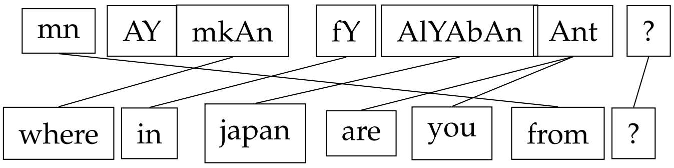


Figure 5.1: A word alignment example of an Arabic to English sentence pair. The Arabic sentence is romanised according to Buckwalter's method.

ferent for different language pairs. Language pairs such as French and English do not need a long distortion span, since they are very similar in their word order differences and most of the reorderings can be captured by the extracted phrases from the bi-text. On the other side, there are language pairs such as Turkish and English with fundamentally different word orders. Turkish is generally a Subject-Object-Verb (SOV) language, which means for many of the sentences a long reordering is required to translate the verb in the right place in the English sentence. However, with a very rich morphology, Turkish word order can vary and some sentences may not need such long distance reorderings.

To improve the phrase-based systems' reordering capabilities, we aim to build a model that scores different reordering decisions based on lexicalised and syntactic features. In addition, we use this model to guide the decoder to dynamically change the size of the reordering window according to the state of translation. Consider the example sentence in Figure 5.1. We want a model to encourage the decoder to skip the first word (`mn`), but translate the next four words monotonically (`mkAn fY AlYAbAn Ant`) and finally jump back to translate the uncovered first word. Thus, we condition our jumps not only on the start and end of the jump, but also on the words jumped over. Additionally, in order to increase the size of the reordering window, we dynamically adjust the distortion limit according the requirements of the reordering model. In other words, the size of the window for hypothesis expansion in the decoder is determined by the current state of the decoder. The latest translated phrase and all the phrases that are about to be translated are taken into account to find the required distortion limit for the next step.

The rest of this chapter is organised as follows: Section 5.1 overviews some of the approaches close to this work. Section 5.2 investigates the importance of distortion parameters in translation quality and speed. Sections 5.3 and 5.4, explain our approach to deal with reordering and Section 5.5 reports experiments done based on the proposed models on several languages. In Section 5.7 onward our participation in IWSLT evaluation campaign is described.

5.1 Related Work

Another category of reordering models, called lexicalised reordering, can be integrated into the decoder as an additional feature or features, so the reordering scores are combined with evidence provided by other features. Lexicalised reordering models were first introduced by [Tillmann, 2004]. They condition the reordering on the previously translated phrase and the next phrase to be translated considering the source and target sides. Different movements are grouped together to deal with data sparsity. [Al-Onaizan and Papineni, 2006] conditioned the exact jumps on the source side words (unigram) and had three features added to the decoder. [Koehn et al., 2005b] considered both source side and target side phrases and predicted three different types of movements of the phrases[2]. In Section 5.3.1, we introduce a new model that in principle is similar to these lexicalised models, however uses the statistics of all the words involved in the jump including those in between.

[Zens and Ney, 2006] proposed a method based on maximum entropy principle to combine different features and predict the word orientation. They combined multiple lexicalised features and for generalisation, considered features based on word classes. They concluded that features based on the source sentence words perform better than those based on the target side and allowing for more context always helps. Since predicting the exact position is not easy, the next positions are grouped together and the model predicts the class of the next jump. Although, they only report the results for a

[2] The model is implemented in the open source SMT system, Moses `http://www.statmt.org/moses`. It is possible to configure the system to build the model with different contexts. For example, only source side or only previous phrases.

small set of classes (backward, monotone and forward), their model is general enough to predict more fine-grained classes. Inspired by their work, [Green et al., 2010] have built two models for each transition. One based on the features of the outbound word (the word that has just been translated) and one model based on features of the inbound word (the word, we are about to translate). Their feature set includes words, part of speech (POS) tags and sentence length features. They argue that using the new models renders the linear future distortion cost inappropriate and add future distortion cost as another feature to be optimised through MERT. In [Xiong et al., 2006] a maximum entropy based model is proposed to predict the orientation of neighbouring blocks in their BTG[3]-based decoder. They have two types of BTG merging rules, straight or inverted and the reordering model weights the merging rules using lexicalised features of the source and target side. Following [Xiong et al., 2008], they extend the model to include linguistically-aware features.

With the same motivation as ours, that different sentence types require different reordering treatments, [Zhang et al., 2008] classify the Chinese sentences under three categories and build reordering models for each category. For sentence type identification, a Support Vector Machine (SVM) classifier is built, with features including all the words in the sentence. They report substantial improvements over the baseline for the Chinese-to-English IWSLT 2007 task.

5.2 Distortion and Translation Quality

As mentioned before, due to the complex nature of decoding in machine translation [Knight, 1999], many parameters are used to manage the size of the search space. Distortion limit or the skip widow size is one of the most important parameters that controls the freedom of the decoder in permuting words to capture the word order differences between the source and the target languages. The best results on different language pairs need different settings for the distortion limit. It is common to set the parameter according to the nature of languages involved and with respect to speed

[3] Bracketing Transduction Grammar

and memory requirements. Longer limits lead the decoder to generate more hypotheses and increase translation time. However, increase in time is not the only drawback of having a longer distortion limit. More hypotheses are generated, therefore more burden is put on the language model to choose the best reordering decision.

Figure 5.2 shows the result of decoding with distortion limits between 1 and 15. Although both graphs show the results of an identical system on two data-sets, the best result for each one of them is achieved by different parameters. One way to find the best distortion limit is to run the tuning process with a range of distortion limits and choose the one with the highest score. Apart from the substantial amount of work required to perform the tuning several times, it is not even guaranteed that the best distortion limit for the development set is the best for the unseen test set.

Another parameter related to distortion is the reordering constraint strategy, which controls the decoder in how to skip words and return back for open positions. [Zens et al., 2004] investigated different reordering constraints and reported their differences on multiple translation tasks. [Dreyer et al., 2007] also proposed a method to find the best reordering constraint independent of other features and solely based on the ability of the constraint to cover all the needed *n*-grams in a sentence. Figure 5.3 shows the translation quality for two different reordering constraints on a Turkish-to-English translation task. One graph of figure 5.3 is constrained by the so-called "Window length" constraint, which restricts the decoder by not letting it to choose a phrase with more than *dl* words distance from the first open position of the source sentence. The constraint in the other graph is "Maximum distortion", which is more relaxed and the only restriction is the distance between the last translated phrase and the next one [Lopez, 2009]. As one can see, the Turkish-English language pair requires relatively long distortion limits, however, the maximum distortion strategy reaches the best results earlier than the window length strategy and overall has a higher score.

We propose a method of selecting the best distortion limit in each step of hypothesis expansion. This method determines the size of the window required to be searched for the next phrase to be translated. Adjusting the distortion limit prevents the decoder to explore undesirable parts of the search space. This saves both time and improves the

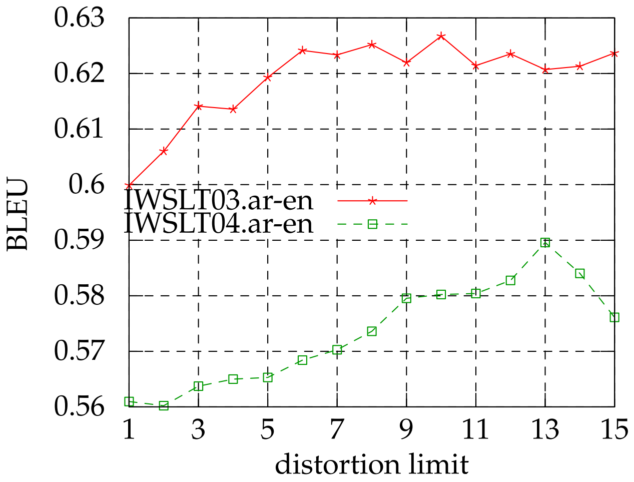


Figure 5.2: The effect of the distortion limit parameter on the quality of the translation system. Both graphs are results of the baseline system (see Section 5.5) on Arabic-English of BTEC task, tuned on IWSLT03.ar-en.

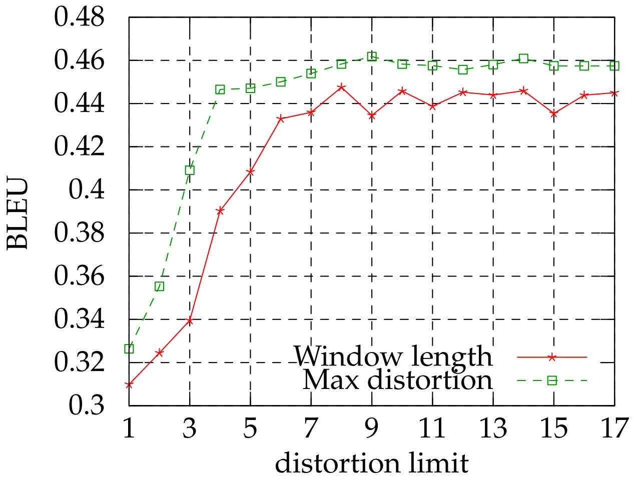


Figure 5.3: Results of two different reordering constraints on the Turkish-English of the BTEC task. Both graphs are the BLEU score of the baseline system on the IWSLT03.tr-en tuning set.

performance by avoiding extra noise during the search. In the next section, we first describe a lexicalised reordering model to establish the main set of features required for a discriminative reordering model.

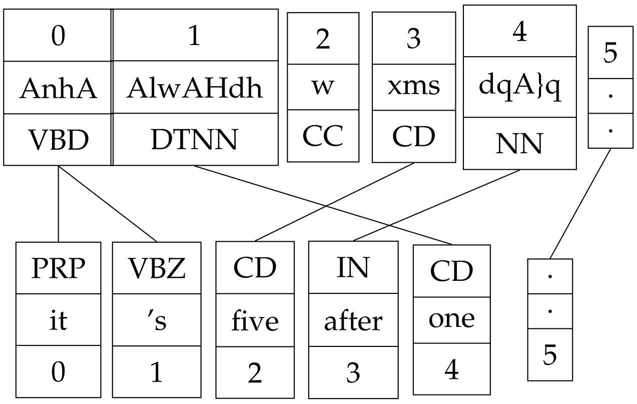


Figure 5.4: A word alignment example of a sentence from the Arabic-English training data. The Arabic sentence is romanised according to Buckwalter's method.

5.3 Reordering Models

5.3.1 Lexicalised Reordering Model

We build a lexicalised reordering model based on [Al-Onaizan and Papineni, 2006] with three additional features modelling the costs of jumping from, jumping to and jumping over the words involved in the reordering. Assume we want to collect training frequencies from the example sentence in figure 5.4. We loop over the target sentence and collect the jump statistics by considering e_i and e_{i+1}, where $0 <= i < I$. For example, for $i = 1$, we consider e_1 and e_2, which are aligned to f_0 and f_3 respectively. The following words are the local context of this jump (from f_0 to f_3) and their respected frequencies will be increased by one:

1. f_0 as outbound word

2. f_3 as inbound word

3. f_1 and f_2 as jumped over words

To avoid collecting evidence for a jumped over word multiple times, the frequency of being jumped over for a position only increases once. We collect the above frequencies for all the jumps in one sentence and all the sentences in the training data.

The training examples defined above will be used to add three additional features to the decoder. As mentioned before, the jumps are binned together to have several jump class. Each class represents a range of jumps, for example, 1 to 4 or 5 to 9. Suppose D is the set of all jump classes and $d_{j,j'}$ is a class associated with a range that the distance between j and j' belongs to. Therefore, $\text{count}_o(f_j, d_{j,j'})$ is the number of times that a jump that belongs to $d_{j,j'}$ is occurred after the word f_j. The outbound feature function that is added to the decoder is:

$$l_o(f_1^J, j, j', d_{j,j'}) = \frac{\text{count}_o(f_j, d_{j,j'})}{\text{count}(f_j)} \tag{5.3.1}$$

which is smoothed by a factor (α) as:

$$l_o(f_1^J, j, j', d_{j,j'}) = \frac{\alpha \frac{\text{count}_o(d_{j,j'})}{\sum_{d \in D} \text{count}_o(d)} + \text{count}_o(f_j, d_{j,j'})}{\text{count}_o(f_j)} \tag{5.3.2}$$

The distance between j and j' is defined as:

$$\text{distance}(j, j') = \begin{cases} j - j' - 1 & \text{if } j \geq j' \\ j' - j & \text{if } j < j' \end{cases} \tag{5.3.3}$$

Two more features l_i (inbound) and l_j (jumped-over), similar to this are also added for inbound and jumped-over words.

We performed a small series of experiments to evaluate the effect of these features on the translation quality and the system equipped with these features improved the baseline (see Section 5.5) for the two best performing distortion limits of the baseline. Table 5.1 shows the results.

5.3.2 Discriminative Reordering Model

The results in the previous section show that the distance-based distortion penalty plus the language model are not enough for making the best reordering decisions. Lexicalised reordering models [Tillmann, 2004; Koehn et al., 2005a] have been shown to

SET	RUN	DL=6	DL=10
IWSLT08(dev)	BASELINE	0.5348	0.5449
	LEX	**0.5461**	**0.5534**
IWSLT07(test)	BASELINE	0.5022	0.5128
	LEX	**0.5121**	**0.5142**

Table 5.1: Comparing the baseline and the lexicalised reordering model with inbound, outbound and jumped-over features. The results are on Arabic-English of BTEC task.

be effective for many language pairs in improving the translation quality. However, because we want to predict the distortion limit, we need to calculate all the reordering costs before decoding the sentence. Additionally, we want to incorporate features extracted from the whole sentence, along with surface features of the phrases we are about to translate in the reordering model. Lexicalised reordering models rely on surface forms of the source and target phrases that have been translated or the ones we are about to translate. Factored models [Hoang and Koehn, 2009] have been proposed to incorporate features such as POS-tags, however, global features such as chunk information are not easily included.

Inspired by [Zens and Ney, 2006], we build a maximum entropy classifier [Berger et al., 1996] that predicts the length of the next jump based on the local lexicalised features and the sentence structure[4].

To increase the classification accuracy, we divide the jumps into a set of classes. For example, jumps with length 2 to 4 are in one class, those with length 5 to 9 in another, etc. Feature functions are binary functions of the form:

$$h_k(f_1^J, j, j', d_{j,j'}) \tag{5.3.4}$$

where, f_1^J is the source sentence with all the syntactic information including POS and chunking tags. h_k is a binary function which is 1 when the feature is present for the specific jump decision and 0, if it is not. j and j' are source positions and $d_{j,j'}$ is the

[4] For a an introduction to maximum entropy principle and the way features are engineered to build the classifier, see 4.3.1.

jump class between them. The decision formula is:

$$p(d_{j,j'}|f_1^J, j, j') = \frac{1}{Z} \exp\,(\sum_{k=1}^{N} \lambda_k h_k(f_1^J, j, j', d_{j,j'})) \tag{5.3.5}$$

where Z is a normalisation factor:

$$Z = \sum_{d \in D} \exp\,(\sum_{k=1}^{N} \lambda_k h_k(f_1^J, j, j', d)) \tag{5.3.6}$$

One of the main benefits of using a discriminative model for this classification task is the ability of these models to learn millions of inter-dependent features. We define an extensive set of features including mostly local context of each jump and some of the characteristics of the sentence. The following list is the set of features used in training the model for a jump from j to j' in sentence f_1^J:

- inbound (IN) and outbound (OUT) words, f_j and $f_{j'}$
- both words together (PAIR), $f_j + f_{j'}$
- jumped over (OVER) words, all the words between j and j' as described in Section 5.3.1
- part of speech tags of inbound, outbound, pairwise and jumped over words (IN.POS, OUT.POS and ...)
- bigram inbound (IN2) and outbound (OUT2), $f_{j-1} + f_j$ and $f_{j'} + f_{j'+1}$
- are both j and j' in the same syntactic chunk or not (1CHUNK and 2CHUNK)?
- does f_1^J contain a question mark (IS.Q)?
- is there a question mark or full stop between j and j' (CROSS.FULL)?
- is there a punctuation mark between j and j' (CROSS.PUNCT)?

Table 5.2 shows the contribution of each set of features to the quality of the model. We used the Arabic-English training data for these experiments. 500 sentences were held-out for validation and 500 sentences were set aside for testing. The rest of the collection was used for training the model.

Features	Accuracy	F_1^M	$\hat{\pi}^M$	$\hat{\rho}^M$
OUT, IN	0.7127	0.5306	0.6538	0.4935
+OVER	0.8337	0.6265	0.7720	0.5874
+PAIR	0.8460	0.6617	0.7940	0.6197
+(*.POS)	0.8826	0.6909	0.8496	0.6503
+(*.POS2)	0.9024	0.7666	0.8392	0.7290
+IS.Q,CROSS.*	0.9042	0.7806	0.8525	0.7404
+IN2,OUT2	0.9085	**0.7964**	0.8643	**0.7566**
ALL	**0.9091**	0.7958	**0.8737**	0.7503

Table 5.2: Classification results of the maximum entropy classifier with different features and the contribution of each set of features. F_1^M, $\hat{\pi}^M$ and $\hat{\rho}^M$ are macro F-measure, macro-precision and macro-recall respectively. macro F-measure is calculated by averaging over the F-measures of each class. *.POS means all the features that their name end with .POS. The evaluation is done on the Arabic-English data set.

5.4 Dynamic Distortion

In Section 5.2, we argued for the importance of determining the optimum distortion limit. Both translation quality and decoding speed are influenced by changing this parameter. The discriminative model described in the previous section, provides us with some information about the reordering needs of a sentence before starting to decode it. This enables us to determine the best distortion limit for this particular sentence and this particular hypothesis expansion.

Changing the distortion limit for each sentence or more specifically for each hypothesis expansion, has a few advantages: Firstly, it removes the need for tuning the system with many different distortion limit settings to find the best one. As it is clear from the results of Section 5.2, the best value for the parameter on one data set may not be the best for another. Secondly, the limit can be very long for some sentences or some parts of a sentence. Changing it for each hypothesis expansion can compensate for long distortion in terms of decoding speed. Basically, we increase the distortion when it is needed and save time when there is no need for long distance reorderings. Thirdly, adjusting the distortion limit reduces the amount of unnecessary jumps in some parts of the sentence and hence decreases noise in the search process, which leads to better translation quality. Additionally, other parameters of the search algorithm that control the size of the search space, such as beam width or stack size can be increased without increasing the decoding time substantially.

Before decoding sentence f_1^J, we use the classifier described in the previous section to compute the probability $p(d_{j,j'}|f_1^J, j, j')$ for each j and j', where $0 <= j, j' <= J+1$ and for all $d \in D$. 0 and $J+1$ are also considered to include the initial move after the start and the final jump before the end symbol. In the next step, the most probable jump after each source position is calculated and the distance is saved as the best distortion limit after that position. To score the jumps after each source position j, equation 5.4.1 is used:

$$s_j(j') = \prod_{j''=j}^{j''=j'} p(d_{j,j''}|f_1^J, j, j'') \prod_{j''=j'+1}^{j''=J+1} (1 - p(d_{j,j''}|f_1^J, j, j'')) \tag{5.4.1}$$

and the distortion limit estimated by this approach for position j equals to:

$$dl(j) = \text{distance}(j, \arg\max_{j'}\{s_j(j')\}) \tag{5.4.2}$$

where distance is defined in equation 5.3.3. This way we find the most likely jump after f_j and set the distortion limit at position j to length of the jump. The above equations are for forward distortion and similar equations are used for backward distortions.

5.5 Experiments

To examine the effects of the discriminative reordering model and the dynamic distortion on translation quality, we have chosen the Arabic-to-English and Turkish-to-English data sets from the IWSLT BTEC task as they involve many short, medium, and long distance reorderings. Some of the statistics of the data sets are shown in Table 5.3.

Data set	**Source lang**	**Sentences**	**Average. len**	**Words**	**Vocabulary**	**OOV**	**Number of refs**
train	Arabic	19972	8.50	169943	14519	-	-
train	Turkish	19972	8.12	162198	6098	-	-
IWSLT03.ar-en	Arabic	506	6.56	3323	1095	111(3.34%)	16
IWSLT04.ar-en	Arabic	500	6.95	3479	1189	101(2.90%)	16
IWSLT05.ar-en	Arabic	506	6.66	3375	1182	124(3.67%)	16
IWSLT07.ar-en	Arabic	489	6.45	3158	1100	165(5.22%)	6
IWSLT08.ar-en	Arabic	507	6.73	3414	1130	153(4.48%)	16
IWSLT03.tr-en	Turkish	506	6.18	3131	1142	152(4.85%)	16
IWSLT04.tr-en	Turkish	500	6.19	3096	1209	175(5.65%)	16

Table 5.3: Corpus statistics and OOV token rates for the development and test sets used for the experiments.

5.5.1 Baseline

The preprocessing stage for Arabic-to-English includes tokenisation of both sides and lower casing of the English side. We removed all the diacritic characters from the Arabic side and normalised punctuation. For tokenising Turkish, we used Morfessor [Creutz and Lagus, 2005] to automatically analyse the morphology of the source side. Lower casing was applied to both source and target sides of Turkish and English.

The decoder is a common multi-beam, multi-stack phrase-based decoder, described in [Koehn et al., 2003] with the following features:

- phrase translation probabilities and lexical probabilities for both directions
- a 4-gram language model
- phrase and word penalties
- distance-based reordering penalty

The weights for the features are optimised by MERT [Och, 2003] to maximise the BLEU [Papineni et al., 2001] score. We optimised the discriminative model using the L-BFGS implementation within the `MALLET` toolkit [McCallum, 2002]. The built model is used to score the reordering options before the decoding.

5.5.2 Results

For the Turkish-to-English task, we tune the baseline (BASELINE) and the discriminative reordering model (DISCRIM-REO) for distortion limits 0 to 17 and tune Arabic-English for distortion limits 0 to 15. For both tasks dynamic distortion method (DYNAMIC-DL) is tuned. Tables 5.4 and 5.5 show the results for the Turkish-to-English and Arabic-to-English tasks, respectively. For both tasks we ran the baseline with the lexicalised reordering model of `Moses` [Koehn et al., 2005b], with no significant improvements, so we did not include the results of the lexicalised reordering model here.

In the Arabic-to-English task the window length constraint performs better than the other constraints. In this constraint the size of the jump is restricted by the first uncovered position of the source sentence. However, since we change the distortion limit during decoding for the dynamic distortion method, an uncovered position outside the window for one move can be inside the distortion limit window for another. Therefore, we relax this restriction in the dynamic distortion method and allow the decoder to make jumps, even if the first uncovered position remains outside the current distortion. Also, we relax the backward distortion limit restriction if there is an uncovered position outside it.

In most cases, DISCRIM-REO performs better than the baseline, particularly on longer distortion limits, which is expected given the fact it has an extra feature to deal with the large amount of reordering decisions. In all the experiments, confirming previous findings [Green et al., 2010], we found that the future distortion cost is crucial for the quality of the translation, particularly for systems with long distortion parameters.

Overall the discriminative model and the dynamic distortion method performed better for Turkish-to-English compared to Arabic-to-English. This can be justified by the fact that Turkish-to-English translation requires more reorderings than Arabic to English.

SET	RUN	DL=6	DL=11	DL=17
IWSLT03(dev)	BASELINE	0.4500	0.4576	0.4574
	DISCRIM-REO	0.4591	**0.4641**	**0.4669**
	DYNAMIC-DL	**0.4640**	0.4640	0.4640
IWSLT04(test)	BASELINE	0.4273	0.4366	0.4363
	DISCRIM-REO	0.4378	0.4434	0.4412
	DYNAMIC-DL	**0.4492**	**0.4492**	**0.4492**

Table 5.4: Experimental results on Turkish-English data sets. The first three rows show the result on the development set and the rest of the results on the test set. set.

SET	RUN	DL=3	DL=6	DL=9	DL=12	DL=15
IWSLT08(dev)	BASELINE	0.5358	0.5348	0.5464	0.5383	0.5416
	DISCRIM-REO	0.5338	0.5458	0.5507	0.5489	0.5489
	DYNAMIC-DL	**0.5571**	**0.5571**	**0.5571**	**0.5571**	**0.5571**
IWSLT03(test)	BASELINE	0.6001	0.6024	0.6199	0.6076	0.6129
	DISCRIM-REO	0.6034	0.6053	0.6220	0.6123	0.6137
	DYNAMIC-DL	**0.6228**	**0.6228**	**0.6228**	**0.6228**	**0.6228**
IWSLT04(test)	BASELINE	0.5619	0.5733	0.5765	0.5789	0.5784
	DISCRIM-REO	0.5534	0.5748	0.5794	0.5820	0.5844
	DYNAMIC-DL	**0.5856**	**0.5856**	**0.5856**	**0.5856**	**0.5856**
IWSLT05(test)	BASELINE	0.5789	0.5875	0.5966	0.5841	0.6007
	DISCRIM-REO	0.5815	0.5922	0.6002	0.5941	0.5853
	DYNAMIC-DL	**0.6016**	**0.6016**	**0.6016**	**0.6016**	**0.6016**
IWSLT07(test)	BASELINE	0.5010	0.5022	0.5103	0.5130	0.5098
	DISCRIM-REO	0.5047	0.5091	0.5196	0.5136	0.5141
	DYNAMIC-DL	**0.5242**	**0.5242**	**0.5242**	**0.5242**	**0.5242**

Table 5.5: Experiment results on Arabic-English data sets. The first three rows show the result on the development set and the rest of the results on the test set.

5.6 Discussion

We showed that choosing the best distortion limit for a language pair or even a data set can gain substantial improvements in phrase-based statistical machine translation decoders. To avoid the difficulty of running with all possible settings, we proposed a method of dynamically adjusting the distortion limit for each hypothesis expansion in phrase-based decoders. To determine the best value for the distortion limit at each move, a discriminative reordering model with numerous features is built and integrated into the decoder as an extra feature.

Results of the experiments by DISCRIM-REO show that more features in the discriminative reordering model helps to improve the accuracy of the classification and the quality of the translation, however, lexical features are more effective than POS or chunk-based features.

Since there is no difference between the features of DISCRIM-REO and DYNAMIC-DL, the improvements achieved by the latter is due to the change of the search space explored by the decoder. Therefore, guiding the decoder during the search can be effec-

Run	English Sentence
Baseline Dynamic-DL Reference	what do you have a newspaper ? what newspaper would you have ? what newspapers do you have ?
Baseline Dynamic-DL Reference	to the city center how much is it ? how much is it to the city center ? how much to downtown ?
Baseline Dynamic-DL Reference	two nights i want to stay . i 'd like to stay for two nights . i 'd like to stay for two nights .
Baseline Dynamic-DL Reference	sales section where can i find it ? where can i find the sales department ? where can i find the sales department ?
Baseline Dynamic-DL Reference	after each meal take this three times a day . take this three times a day after meals . take it three times a day after meals .

Figure 5.5: A few examples of translation by the baseline and the dynamic distortion method.

tive in improving the quality of translation.

5.7 The System Description for an Evaluation Campaign

We as the QMUL[5] team submitted runs at IWSLT 2010[6] evaluation campaign for all the three language pairs of BTEC task. The BTEC standard translation task focuses on frequently used utterances in the domain of travel conversations [Paul et al., 2010]. In 2010, the translation task was provided for the translation of Arabic, French and Turkish spoken language text into English.

This section reports the technical details of the system used to perform the translation and the particular improvements of the baseline system to make our submission more competitive.

Our main focus in this submission was on improving the reordering capabilities of the decoder, however, improvements were gained by experimenting with different word-

[5] Queen Mary, University of London

[6] The 7th International Workshop on Spoken Language Translation, `iwslt2010.fbk.eu`

alignment strategies and dealing with out of vocabulary (OOV) words.

The training data provided for the IWSLT BTEC task was relatively small and since the sentences are transcripts of conversations, most of them are very short. This enabled us, to perform the cycle of training, tuning and testing more frequently and investigate many small features and changes. A few of the modifications helped the translation performance, while most of them had insignificant impact.

5.7.1 Baseline System

Preprocessing

For the Arabic-English task, we removed all the diacritics from the Arabic side and normalised the numbers and the punctuations. Buckwalter's morphological analyser is used to tokenise the Arabic side and a simple English tokeniser and lower-caser for the English side.

For French-English pair, we used a simple tokeniser, which works for all European languages in addition to lowercasing both sides. It separates most of the words by whitespace and punctuation characters, but keeps a few exceptions based on a manually created list.

For Turkish-English pair, we used Morfessor [Creutz and Lagus, 2005] to tokenise the Turkish side. Morfessor finds segmentation of the words in an unsupervised manner. The Turkish side of the bitext and all the development data are fed into the Morfessor algorithm to produce segmentations for words which often are similar to linguistic morphemes. Morfessor divides words into multiple morphs including prefixes, stems and suffixes. We retain all the morphs and separate them by a whitespace. We avoided using other publicly available Turkish morphological analysers, since they were using extra training data. We lower-cased both sides of this language pair. Table 5.6 shows the effect of the preprocessing step on the vocabulary size of the data sets.

	Arabic	**French**	**Turkish**	**English**
Tokens w/o tokenisation	159k	160k	112k	153k
Tokens w tokenisation	170k	200k	162k	189k
Vocabulary w/o tokenisation	37516	35799	39545	32619
Vocabulary w tokenisation	14519	9212	6098	7182
Singletons w/o tokenisation	29852	28572	32410	26444
Singletons w tokenisation	7426	4232	711	3116

Table 5.6: The effect of preprocessing on the number of tokens and the vocabulary size for all three language pairs. Singletons are words that occur once in the collection.

Data set	**Source**	**Words**	**Vocabulary**	**OOV before**	**OOV after**
IWSLT03.ar-en	Arabic	3323	1095	111	64
IWSLT04.ar-en	Arabic	3479	1189	101	47
IWSLT05.ar-en	Arabic	3375	1182	124	56
IWSLT07.ar-en	Arabic	3158	1100	165	78
IWSLT08.ar-en	Arabic	3414	1130	153	77
IWSLT09.ar-en	Arabic	3135	1039	155	82
IWSLT10.ar-en	Arabic	3207	1096	127	54
IWSLT03.fr-en	French	4063	957	92	69
IWSLT04.fr-en	French	4068	1026	85	52
IWSLT05.fr-en	French	4052	994	89	65
IWSLT09.fr-en	French	3877	888	70	45
IWSLT10.fr-en	French	3813	901	61	43
IWSLT03.tr-en	Turkish	3131	1142	152	86
IWSLT04.tr-en	Turkish	3096	1209	175	89
IWSLT09.tr-en	Turkish	2944	1071	137	79
IWSLT10.tr-en	Turkish	2910	1102	125	76

Table 5.7: Number of OOV tokens in the development set before finding replacements and after.

Out-of-Vocabulary Words

For a small size training data such as the one provided, unknown words are a significant problem. Intuitively, many of the unknown words are morphological variations of known words, particularly for morphologically rich languages such as Arabic and Turkish. Therefore, we used simple stemming algorithms to find matches of the unknown words. We search to find a match for the unknown word in the test data among the stemmed words in the training data, then we look for finding a match for the stemmed version of the unknown words in the original training data. Finally, the search is done to find a match of the stemmed unknown words in the stemmed training data. For any match found, the unknown word is replaced with the unstemmed word in the training data. Table 5.7 shows the number of OOV tokens before and after the replacement.

Decoder

The features of the baseline include:

- phrase translation probabilities and lexical probabilities for both directions. The word alignment models were produced using Berkeley Aligner [Liang et al., 2006]. For all three language pairs, we ran IBM model 1, IBM model 2 and HMM jointly for 5 iterations.
- a 4-gram language model. SRILM toolkit [Stolcke, 2002], was used to build a 4-gram language model, which includes all 4-grams. SRILM by default excludes 4-grams that occur only once in the training data. Preliminary experiments showed that including them improves the quality of the translation for the three language pairs.
- phrase and word penalties.
- distance-based reordering penalty.

There are two distortion parameters in our decoder. The distortion limit, which determines the window size of the reordering and the distortion constraint, which controls the decoder movement mainly based on the first uncovered position. Figures 5.6 and 5.7 show the BLEU score for different values of the distortion limit for Arabic-English and French-English. The best distortion limit for Arabic in average is 13, which is not the best performing on the tuning data. In other words, the best distortion limit chosen based on the tuning data is not the best for the testing data. Figure 5.3 shows the BLEU score for Turkish-English with two different distortion constraints. One is the so-called "Window" constraint [Lopez, 2009] and the other is called "Max distortion" [Moore and Quirk, 2007]. As mentioned before, the window constraint restricts the decoder by not letting it choose a phrase with more than *dl* words away from the first open position of the source sentence, while the max distortion constraint is relaxed about the first open position and only restricts the decoder to select the next phrase in a window of length 2 × *dl*. For all experiments the future distortion cost was also estimated and showed to be crucial, particularly for long distance reorderings.

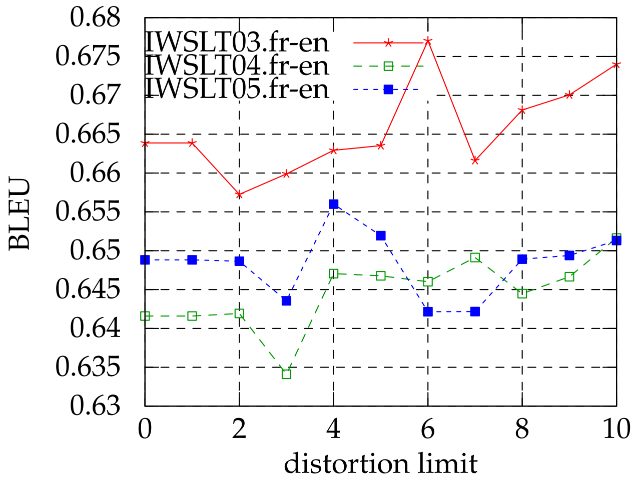


Figure 5.6: BLEU score changes with different distortion limit values for French-English language pair. IWSLT03 is used for tuning and the rest for testing.

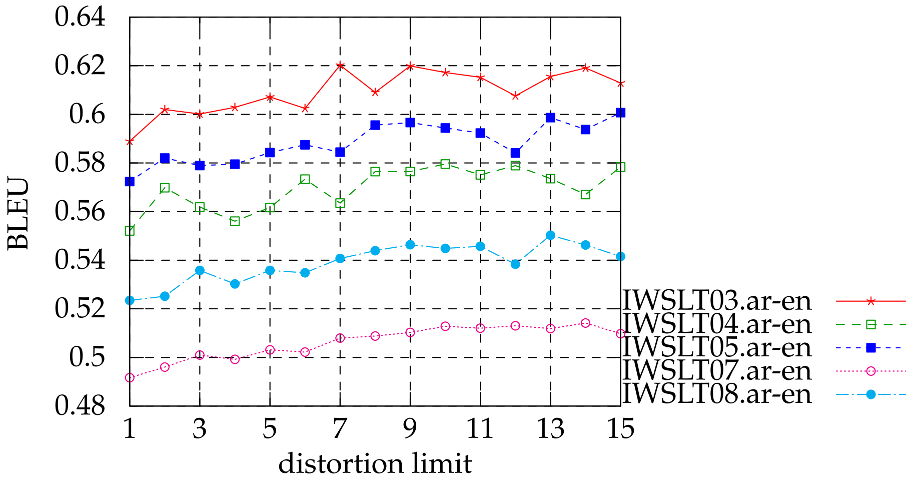


Figure 5.7: BLEU score changes with different distortion limit values for Arabic-English language pair. The graph in the bottom (IWSLT08) is used for tuning and the rest for testing.

The decoder was tuned using Minimum Error Rate Training [Och, 2003], implemented in ZMERT [Zaidan, 2009] to maximise BLEU [Papineni et al., 2001].

Post-Processing

The final output of the decoder was generated through Minimum Bayes Risk Decoding [Kumar and Byrne, 2004], which produced a small, but consistent improvement for all the language pairs. We built a true-caser language model based on the target side of the training data to predict the words that need to be cased. In addition, a detokeniser is used to reverse the tokenisation process.

5.7.2 Reordering Models

The discriminative reordering model of Section 5.3.2 and the dynamic distortion method of Section 5.4 were used to help the reordering capabilities of the decoder. In the discriminative reordering model, the jumps are divided into classes to increase the classification accuracy. For example, jumps with length 2 to 4 are in one class, those with length 5 to 9 in another, and so on .

The set of features that we used for the reordering model include lexicalised words, POS-tags, chunks and sentence type. Features for a jump from j to j' in a sentence f_1^J are:

- $f_j, f_{j'}, f_j + f_{j'}$
- all the words between j and j'
- part of speech tags of the above words: POS(f_j), POS(f_j'), . . .
- bigrams: $f_{j-1} + f_j$ and $f_{j'} + f_{j'+1}$
- bigram part of speech tags of j, j' and the words between them.
- a binary feature indicating that both j and j' are in the same syntactic chunk or not?
- binary feature indicating that f_1^J contains a question mark or not?

- is there a question mark or full stop between j and j'?
- is there a punctuation mark between j and j'?

For Arabic-English and French-English tasks we used all the above features, but for Turkish-English, since we used Morfessor to tokenise the Turkish side, the part of speech and chunking features were excluded.

The classifier was optimised by the L-BFGS method [Nocedal, 1980], implemented in MALLET [McCallum, 2002]. To prevent over-fitting, L_1 regularisation was used to reduce the complexity of the model, however, lower translation performance was achieved by using the regularisation. The regularisation can be viewed as a method to select important features and it improves the classification performance of the reordering model in our experiments, but it leads to the translation performance loss at the end.

5.7.3 Experiments

To find the best setting to translate the final test files, we tune the system on different data sets and tested it on the rest of the data sets and chose the data set for tuning with more consistent improvements. Tables 5.8, 5.9 and 5.10 show results for baseline alone, with the OOV replacements and with the dynamic distortion method. No post-processing, as defined in Section 5.7.1, was applied for the results of the dev data, hence, BLEU scores are calculated on the unprocessed output of the decoder.

To evaluate the contribution of each feature in the classification performance of the discriminative reordering model, we started with the lexical features of f_j and f'_j and added all the features described in Section 5.3.2 one by one. The most substantial improvements achieved by adding the following features:

- all the words between j and j', which is a binary feature indicating the presence of a word between j and j' or not.
- $f_j + f_{j'}$, which indicates the occurrence of f_j and f'_j together.

- bigram part of speech tags of f_j, f'_j and the words between them. For example, POS(f_{j-1})+POS(f_j)

As mentioned before, the part of speech and chunk features were only used in building the models for Arabic-English and French-English language pairs. For Turkish-English, we only used features that did not require part of speech and chunking information.

PRIMARY runs are the baseline with the dynamic distortion method, replacements of the unknown words and post-processing.

SET	RUN	BLEU
IWSLT08(dev)	BASELINE	**0.5821**
	+OOV-REP	0.5751
	+DYNAMIC-DL	0.5754
IWSLT04(test)	BASELINE	0.5993
	+OOV-REP	0.5982
	+DYNAMIC-DL	**0.6018**
IWSLT05(test)	BASELINE	0.6133
	+OOV-REP	0.6157
	+DYNAMIC-DL	**0.6187**
IWSLT07(test)	BASELINE	**0.5383**
	+OOV-REP	0.5357
	+DYNAMIC-DL	0.5351
IWSLT09(test)	PRIMARY	0.5276
IWSLT10(test)	PRIMARY	0.4425

Table 5.8: BLEU scores on Arabic-English data sets. OOV-REP is the baseline with some of the unknown words replaced by the matched known word. DYNAMIC-DL is the baseline with the discriminative reordering model and the dynamic distortion method.

In some of the experiments, the BLEU score decreased after replacing the unknown words with the stemmed matched known words. However, by manually checking the matches, most of the them were good replacements and contributed to the meaning of the sentence, therefore, we included this feature for the final tests.

5.7.4 Evaluation Campaign Results

In total, 20 research groups participated in the three BTEC translation tasks, submitting 12 runs for Arabic-English, 9 for French-English and 8 runs for Turkish-English [Paul

Set	Run	BLEU
IWSLT03(dev)	Baseline	0.6860
	+OOV-Rep	0.6834
	+Dynamic-DL	**0.6874**
IWSLT04(test)	Baseline	0.6605
	+OOV-Rep	0.6630
	+Dynamic-DL	**0.6694**
IWSLT05(test)	Baseline	0.6650
	+OOV-Rep	0.6600
	+Dynamic-DL	**0.6668**
IWSLT09(test)	Primary	0.6180
IWSLT10(test)	Primary	0.5362

Table 5.9: BLEU scores on French-English data sets. OOV-Rep is the baseline with some of the unknown words replaced by the matched known word. Dynamic-DL is the baseline with the discriminative reordering model and the dynamic distortion method.

et al., 2010]. Tables 5.11 , 5.12 and 5.13 show the rank of our submission for each language and for the two data sets provided, namely `IWSLT10(test)` and `IWSLT09(test)`[7]. For the full results and the ranks of other participants, see [Paul et al., 2010].

The results of this evaluation campaign, particularly French to English, were significant for several reasons: Firstly, it was a substantially better performance for QMUL team compared to previous evaluation campaigns. Secondly, it showed the viability of the dynamic distortion method in competing with other well performing systems. Thirdly, the results of our submission were achieved without using system combination or language specific techniques for improving the results. In other words, the dynamic distortion method was the only extra feature of over system compared to a common system that can be built by available open source tools.

Summary

In this chapter, we developed a reordering model that takes into account several features including bigrams, part-of-speech tags and sentence punctuations to predict the reordering requirements of the next phrase expansion. In addition to the reordering

[7] `testset_IWSLT10` and `testset_IWSLT09(test)` in the overview paper.

Set	Run	BLEU
IWSLT03(dev)	Baseline	0.4783
	+OOV-Rep	0.4797
	+Dynamic-DL	**0.4814**
IWSLT04(test)	Baseline	0.4507
	+OOV-Rep	0.4505
	+Dynamic-DL	**0.4577**
IWSLT09(test)	Primary	0.5354
IWSLT10(test)	Primary	0.5128

Table 5.10: BLEU scores on Turkish-English data sets. OOV-Rep is the baseline with some of the unknown words replaced by the matched known word. Dynamic-DL is the baseline with the discriminative reordering model and the dynamic distortion method.

A	BLEU	METEOR	TER	NIST	**z-avg**
IWSLT10(test)	5	5	5	5	5
IWSLT09(test)	4	5	5	3	4

Table 5.11: The rank of our submitted system for the Arabic-English language pair. 12 systems submitted runs for this language pair.

F	BLEU	METEOR	TER	NIST	**z-avg**
IWSLT10(test)	2	8	5	8	8
IWSLT09(test)	4	6	5	8	5

Table 5.12: The rank of our submitted system for the French-English language pair. 9 systems submitted runs for this language pair.

T	BLEU	METEOR	TER	NIST	**z-avg**
IWSLT10(test)	4	5	5	6	5
IWSLT09(test)	4	7	6	7	6

Table 5.13: The rank of our submitted system for the Turkish-English language pair. 8 systems submitted runs for this language pair.

model, we extended the decoder to dynamically adjust the distortion parameter and make long distance jumps possible by avoiding reordering in other parts of the sentence. Experiments on different language pairs were carried out and results showed that the even though reordering model is beneficial, while accompanied by the dynamic distortion technique achieves the best results.

CHAPTER 6

Evaluation of Named Entity Recognition on Statistical Machine Translation Output

In the two previous chapters, we proposed and discussed reordering approaches for improving the translation quality with respect to automatic evaluation metrics. In this chapter, on the other hand, we focus on evaluating statistical machine translation in general and the effect of reordering in particular with respect to specific natural language processing tasks. In other words, we aim to investigate the viability of SMT in performing multi-lingual tasks of natural language processing. This chapter deals with the evaluation of *Named Entity Recognition* on the MT output and the next chapter focuses on text fragment alignment and cross-lingual similarity estimation.

The rest of this chapter is organised as follows: Section 6.1 discusses the rationale behind using MT to solve multi-lingual NLP tasks and Section 6.2 overviews the literature for related work. Section 6.3 gives a brief introduction to named entity recognition and the algorithms used in this work to perform NER. In Section 6.4, we describe the method of evaluation of named entity recognition on machine translation output and Section 6.5 reports the experiments.

6.1 Using MT to Perform Multi-Lingual NLP

With the success of statistical machine translation, a promising way of solving multilingual and cross-lingual natural language processing tasks is to translate the text into a language with more sophisticated tools available (mostly English) and perform the task on the translated text. Since the quality of even the best SMT systems differs for different language pairs and heavily depends on the available language resources, it is important to evaluate the performance of different NLP tasks on machine translation output.

On the other hand, although automatic machine translation evaluation metrics are essential in developing machine translation systems, they do not always reflect the quality of different systems against each other. There are several automatic evaluation metrics [Papineni et al., 2001; Lavie and Agarwal, 2007], however, the correlation of their ranking against human judgement is not always good enough [Callison-Burch et al., 2006]. In addition, the automatic metrics are measuring the overall quality of machine translation output and are used to compare different systems. Surely for some tasks the overall quality of the output is not as important as other aspects of the translation or the translation is good enough even though there are fluency or grammatical problems. For instance, machine translation systems are used to make it possible for the user to understand the idea behind a text and not all the details. There are different levels of understanding of the text and for various tasks the focus is on different aspects. The author's intention, the main entities of the text, the relationship between the entities or time, date and order of events can be the main focus of the reader.

Named entities or noun phrases are content-wise among the most important structures of a text. Correctly detecting and classifying them can be crucial in understanding the text and is beneficial for other natural language processing tasks. In this work, we evaluate the named entity recognition and classification algorithms on machine translation output to investigate the feasibility of using MT systems for named entity recognition and also the effect of machine translation quality, particularly reordering, on the quality of named entity recognition. It is important to improve NE recognition quality on the target language of machine translation, since there are already high quality NER models for languages such as English. Building language dependent NE recognisers for new languages is a labour-intensive and language independent algorithms also need to be adapted for each source language.

The translation process can affect the quality of extracting named entities by incorrectly translating some words or distorting the context which the entity occurs in. Therefore, an effective reordering model that captures the words movements well, has the potential to substantially improve the named entity extraction.

To evaluate NER on MT output, we have used the performance of NER classifiers on the reference translations as an upper bound and gold standard. Test collections with human annotation for a wide range of domains and languages were not available, therefore even though the automatic NE extraction on the reference translations is not perfect, it is a viable choice to estimate the relative performance of NER on the generated output.

6.2 Related Work

Although the main method of evaluating machine translation is human assessment, automatic metrics are of great value in developing new methods and quickly comparing a large number of systems. Due to complexity of MT evaluation, a wide range of evaluation measures have been proposed, which neither of them can be applied in all circumstances. The FEMTI framework has been proposed by [King et al., 2003] to combine different aspects of MT evaluation and provide a way to adjust the evaluation to user needs. On the other hand, there has been efforts to measure some aspects of translation problem and use it to evaluate the effectiveness of a method in that respect. [Birch et al., 2010] propose a method to measure the quality of translation with regard to word order choices. A reordering performed for translating a sentence can be encoded as a permutation and permutation distance metrics are used to quantify the reordering decisions made during the translation. In another work [Birch and Osborne, 2010], they merge the reordering metric with a lexical metric such as BLEU [Papineni et al., 2001] to provide a combined metric to measure both reordering and word choices.

The effect of translation on ad-hoc retrieval is investigated in [Dolamic and Savoy, 2010]. Queries in different languages are translated to English via two commercial machine translation systems and run against a collection of English documents. Four parametric and non-parametric retrieval models are tested and results are compared to a mono-lingual run. The experiments show that the quality of translation directly affects the retrieval performance across different languages, for all the retrieval models and in general the translation process degrades the retrieval quality compared to mono-lingual retrieval.

Named entity translation has been of interest for a long time [Al-Onaizan and Knight, 2002; Koehn, 2003; Huang, 2005] and translation techniques are developed to deal with noun phrases and named entities. On the other hand, automatically acquired parallel named entities are used to improve rule-based machine translation [Toral and Way, 2011]. A lexicon of named entities in several languages are built using data acquired from Wikipedia. The entities are extracted and scored based on their number of occurrences in the corpus. The resulting lexicon is inserted into the rule-based machine translation dictionary and it is shown to substantially improve the translation quality for some language pairs.

A close research to this work is [Babych and Hartley, 2004], which evaluates the performance of ANNIE named entity recognition module in GATE [Cunningham et al., 2002] on several machine translation systems including rule-based and statistical. They conclude that automatic metrics and even human evaluations can not reliably predict the performance of a named entity extraction system on translation output, however, in general higher quality translation systems are more likely to have better quality named entity extraction. In another work, [Babych and Hartley, 2008] investigate the sensitivity of BLEU versus the quality of named entity recognition as a task-based evaluation metric, where the quality of translation is high. The results show that BLEU is not as able as the task-based metric to distinguish between high quality MT systems. They argue that BLEU measures the translation quality in the lexical level. Therefore in the case of high quality MT systems with similar capability of resolving lexical problems, it becomes more difficult to discriminate between them, while a task-based metric evaluates the differences in higher levels such as long-distance syntactic agreement.

In the next section, a brief overview of named entity recognition and common methods of performing it are presented.

6.3 Named Entity Recognition

An essential task of any information extraction procedure is named entity recognition, which is defined as finding spans of text that constitute proper names. Named entity recognition and classification is concerned with finding entities such as people names, organisations and also nonentities such as temporal and numerical expressions in the text and classifying them under their correct categories. Several entity type hierarchies have been proposed [Brunstein, 2002] and specialised domains define specific sets of types for the entities. Meanwhile, many evaluation campaigns expect categorisation of entities under four general categories of PERSON, ORGANISATION, LOCATION and MISC. Therefore, in this work we evaluate all the runs on these general categories.

The problem of named entity recognition is twofold. First finding a span of the text which is an entity and then classifying it under one of the categories. So, the first step is to classify each word under two categories[1] of inside the span or outside the span and then classifying them under the categories. There are many examples that a span of text can be classified under different types and there is an ambiguity, which should be resolved based on the context. There are multiple sources of evidence, including labels of the words before the current word, that must be taken into account for making a decision about each token in the text. To incorporate multiple evidence and the outcome of earlier classifications most of the times named entity recognition is performed as a word sequence labelling task. In this approach, tokens are labelled by classifiers trained on manually annotated data. An important part of building the classifier is selecting a set of features to represent the text, which is suitable for named entity recognition. Features commonly used in training NER systems include [Nadeau

[1] Each word is mostly classified under three categories: the first word inside the span, inside the span and outside the span.

and Sekine, 2007]: word-level features such as the token to be labelled, shape of the token, suffixes, part-of-speech tag of the word and patterns of the word [Collins, 2002], gazetteers and lists features that indicate the presence of the word in a named entity list, and context features that consider words or *n*-grams surrounding the word. As it is expected, many of these features depend on the surrounding of the word that is being classified. An incorrect choice of word order in translation not only degrades the quality of choosing the right translations, it affects the context and surrounding words that is crucial for named entity recognition.

In Section 6.5.1, the two named entity recognition systems used in the experiments of this work and their use of different features have been described in more detail.

6.4 Evaluation Method

To be able to thoroughly evaluate the performance of named entity recognition algorithms on statistical machine translation output, we need multiple test sets and different language pairs. Since large test sets of manually annotated cross-lingual data for many languages are not available, we evaluate the quality of annotation against the quality of the same NER algorithm on the translation reference data. In other words, the judgement data are produced by performing named entity recognition on the reference translations. The evaluation method, estimates the quality of the named entity recogniser compared to the output of the same algorithm on human translated sentences.

Although the quality of named entity recognition in general, depends on the ability of the NER algorithm in detecting spans of text and correctly classifying them, our evaluation method does not depend on the quality of the NER algorithm, because the same algorithm is run on the reference translations and the machine translated sentences. However, the ability of a named-entity recogniser in working with imperfect sentences can reduce the difference in quality between two runs with different translation qualities.

Figure 6.1, sketches the algorithm used to evaluate the performance of a named entity recogniser on translation output of a test set with one reference translation.

```
Input: $f$ and $e$  // $f$ is a foreign sentence and $e$ is the reference translation in English
$e' \leftarrow$ translate$(f)$  // translates the foreign sentence and store it in $e'$
$C \leftarrow$ chunk$(e)$  // extracting named-entities and storing the set of extracted named-entities in $C$
$C' \leftarrow$ chunk$(e')$
$a \leftarrow$ word-align$(e, e')$  // aligning the words in the reference and the translation
for all $c'_i$ in $C'$ do
  for all $c_j$ in C do
    if is-aligned$(a, c'_i, c_j)$ then  // if the words in $c'_i$ and $c_j$ are aligned in $a$
      $tp \leftarrow tp + 1$  // increment the number of true positives
    end if
  end for
end for
$fp \leftarrow |C'| - tp$  // $|X|$ is the size of set $X$
$fn \leftarrow |C| - tp$
```

Algorithm 6.1: Evaluation algorithm for runs with one reference translation. tp (true-positives), fp (false-positives) and fn (false-negatives) are used for calculating precision and recall.

After translating the foreign sentence f into e', the named entity recogniser is run on e' and the reference translation e to produce C' and C respectively, where C, is the set of named entities extracted from e. To find the corresponding named entities, a mono-lingual word aligner is run to link each word in e to at most one word in e'. The word aligner is adopted from [Banerjee and Lavie, 2005], which uses multiple modules to match the words. Exact module to match words that are exactly the same, stem module that matches words which are the same after stemming and WordNet module that matches words that are synonyms according to WordNet. The next step is to find a match for each named entity in C' with an entity in C. Two entities are matched if the words inside them are not aligned to outside words and there is a link between at least two words inside them. If a match is found for an entity in C', then the number of true positives are incremented by one. Since we want to evaluate the performance of named entity recognisers in both aspects of detecting named entities and classifying them, two sets of statistics are collected for each test set and two types of matching are defined. Firstly, a typeless match which is a match between two entities that are aligned based on the word alignment data regardless of their type. Secondly, a typed match which is similar to the typeless match with respect to the alignment data with additional type equality constraint.

Having calculated the number of true positives, the number of false positives is calculated as the number of extracted named entities from the translated sentence $|C'|$, minus the number of true positives. The number of false negatives is equal to the number of extracted named entities from the reference sentence $|C|$, minus the number of true positives. The number of true negatives is not relevant for this problem and is not required for calculating precision and recall. In the case of test sets with multiple reference translations e^*, the same algorithm is run for each (e', e) pair, where $e \in e^*$. The reference sentence that has the highest number of match with the translation is selected and considered to calculate all the statistics.

After calculating tp (true-positives), fp (false-positives) and fn (false-negatives), precision, recall and F-measure are calculated for each sentence and for the entire test set. For all three metrics macro and micro methods[2] are computed:

$$\text{precision} = \frac{tp}{tp + fp} \tag{6.4.1}$$

$$\text{recall} = \frac{tp}{tp + fn} \tag{6.4.2}$$

$$f_{\beta=1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6.4.3}$$

6.5 Experiments

Several language pairs are selected to evaluate the named entity recognition algorithms on machine translation output. The target language of all of them is English, since the best named entity recognisers are written for English. The language pairs and the data collections used for the experiments are as follows:

[2] For description and details of computing macro and micro F-measure, precision and recall, please refer to [Sebastiani, 2002]

- Arabic-English: Arabic has a different word order than English, there are sentences that start with verb (VSO) and noun modifiers follow the noun. Overall, there are many medium range reorderings involved in translating Arabic. The writing system differs from English, which makes replacing the OOV words with the source word not useful at all.

- French-English: French is very similar to English in terms of word order and there are mostly local reorderings of noun modifiers. The writing systems are almost identical except for letters with diacritics, therefore replacing unknown words with source words particularly for proper names can be beneficial.

- Turkish-English: Turkish translation requires short and long-distance reordering and the writing system is different from English in a few letters of the alphabet.

- German-English: German has a very similar writing system to English, but long-distance reordering is required in many of the sentences.

- Bulgarian-English: Bulgarian has an almost entirely different alphabet from English, but is very similar in word order to English.

- Greek-English: Although the main word order of Greek is SVO, other word orders are very common, so the amount of reordering in translation from Greek depends on the test set. The writing system is completely different from English and replacing the unknown words with their source is unlikely to be helpful.

Since not all sentences contain named entities, for each language pair we have concatenated a few test sets to make the number of sentences with named entity large enough for reliable evaluation. The Arabic, French and Turkish test sets are concatenation of IWSLT 2010 [Paul et al., 2010] development data. The German test set consists of concatenation of Europarl and news commentary development data provided for WMT 2010 [Callison-Burch et al., 2010], and for Bulgarian and Greek, 8,000 sentences were set aside from the Europarl corpus [Koehn, 2005], while the rest of the corpus is used for training the translation, language and reordering models. Some statistics of the test sets are provided in Table 6.1.

Language pair	**Data set**	**Sentences**	**Average. len**	**Words**	**OOV**	**Number of refs**
Arabic-English	IWSLT 2010	2,508	6.57	15,204	18%	6
French-English	IWSLT 2010	1,512	6.50	9,193	13%	16
Turkish-English	IWSLT 2010	1,006	6.03	6,076	17%	16
German-English	WMT 2009-10	14,582	24.59	358,663	14%	1
Bulgarian-English	Europarl v6	7,989	23.69	17,447	6%	1
Greek-English	Europarl v6	8,000	25.35	202,800	4%	1

Table 6.1: Test sets statistics. OOV indicates the out-of-vocabulary rate of words in the test sets.

For translation of all languages, the source sentences were tokenised and lower-cased and because the named entity recognisers use their own tokenisation algorithms the output of the translations were detokenised and true-cased. The true-casing models were built based on the target side of the parallel corpus.

6.5.1 Translation Setup

The statistical machine translation decoder is the baseline decoder that its main components were described in Chapter 2. The features used in the decoder for the experiments are:

- phrase translation probabilities and lexical probabilities for both directions
- a 4-gram language model
- phrase and word penalties
- distance-based re-ordering penalty

The weights for the features are optimised by MERT [Och, 2003] to maximise the BLEU [Papineni et al., 2001] score. Apart from the baseline runs, we ran the decoder with additional reordering features to see the change in the quality of translation and the extraction of named entities. The reordering model is the dynamic distortion model that we proposed in the previous chapter and it includes a discriminative reordering model based on several features, including part-of-speech, type of the sentence, number of crossed punctuations and others. This model is explained in detail in [Yahyaei and Monz, 2010a].

6.5.2 Named Entity Recognition Setup

Two named entity recognisers are used to perform the evaluation. Firstly, LingPipe 4.0.1 [Alias-i, 2010], which is an HMM chunker with the ability to tag tokens as the beginning, in the middle or the end of a chunk with a specific type. The chunker produces an n-best chunking candidates and a model based on character language models is used to rescore the n-bets list [Carpenter, 2006]. For the experiments in this work we have used a model trained on the news data provided in MUC-6 which accompanies the LingPipe distribution.

The second named entity recogniser is Stanford NER, which is one of the best performing open source NERs available [Finkel et al., 2005]. Stanford NER is a sequence labelling classifier based on conditional random fields. The model is trained with a lot of features such as lexical features including the current word, previous word and next word, orthographic features, prefixes and suffixes, label sequences and so on. In addition, based on a large unannotated data words are clustered and used as extra features. The classifier is trained on a mixture of several named entity corpora, which makes it robust across domains.

6.5.3 Results

To tune the distortion parameters of the decoder for each test set, we ran the decoder on the tuning set for different distortion limits in the range of 0 to 20 and picked a distortion parameter with the highest development BLEU. Tables 6.2 and 6.3 show the results of named entity recognition on the baseline and discriminative reordering model translations. For each test set macro and micro, F-measure, precision and recall are shown in the tables. The BLEU scores reported here are the result of evaluating the first test set for each language pair against all the reference translations.

The Turkish, French and Arabic test sets consist of spoken short sentences which mostly do not contain named entities. The improvements in extracting named entities for these languages are not substantial, which in the case of French and Arabic can be explained by the fact that the reordering model has not achieved a significant better quality compared to the baseline. However, the best distortion parameters for all three languages, which are 13, 6 and 12 for Turkish, French and Arabic respectively, indicate the importance of reordering. Although the discriminative reordering model improves the quality of translation for the Turkish test set, both named entity systems perform better on the baseline translations. One reason for this can be the fact that Turkish word order is different from English, but the reordering situations which have been

Language	Run	BLEU	MF_1	μF_1	MP	μP	MR	μR
Turkish	Baseline	45.80	51.27	74.26	53.82	83.93	53.71	70.15
	Disc-Re	46.46	48.61	71.97	49.03	81.90	49.52	64.18
French	Baseline	64.60	51.36	79.25	51.67	87.50	51.99	72.41
	Disc-Re	64.92	51.99	80.00	52.53	88.73	52.45	72.83
Arabic	Baseline	60.76	51.43	70.17	52.04	83.92	52.09	60.29
	Disc-Re	60.80	50.99	70.36	51.74	85.05	51.38	60.00
Bulgarian	Baseline	42.63	79.55	80.02	78.91	78.19	82.68	81.94
	Disc-Re	42.34	79.87	80.20	79.25	78.32	83.01	82.18
Greek	Baseline	42.72	25.17	29.68	28.74	41.99	28.07	22.94
	Disc-Re	43.08	25.19	29.67	28.70	41.99	28.09	22.94
German	Baseline	26.14	68.61	69.83	70.20	71.93	69.63	67.85
	Disc-Re	27.34	69.50	70.65	71.00	72.52	70.46	68.88

Table 6.2: Results of the Stanford NER system regardless of entity type, where M stands for macro and μ stands for micro. F_1 is F-measure, P is precision and R is recall.

Language	Run	BLEU	MF_1	μF_1	MP	μP	MR	μR
Turkish	Baseline	45.80	42.96	65.23	43.49	67.35	43.53	63.24
	Disc-Re	46.46	42.48	65.58	43.26	68.06	42.84	63.28
French	Baseline	64.60	44.12	70.75	44.76	74.77	44.44	67.14
	Disc-Re	64.92	43.67	70.37	44.46	75.20	43.77	66.13
Arabic	Baseline	60.76	43.33	57.60	43.71	58.67	44.26	56.57
	Disc-Re	60.80	43.25	58.15	43.67	59.51	44.15	56.86
Bulgarian	Baseline	42.63	66.67	70.42	67.02	70.85	69.00	70.00
	Disc-Re	42.34	66.66	70.35	67.06	70.81	68.97	69.88
Greek	Baseline	42.72	25.48	28.53	29.92	35.20	26.43	23.99
	Disc-Re	43.08	25.29	28.40	29.82	34.97	26.22	23.90
German	Baseline	26.14	50.87	54.65	52.08	56.02	52.83	53.34
	Disc-Re	27.34	52.04	55.82	53.36	57.30	53.78	54.41

Table 6.3: Results of the LingPipe NER system regardless of entity type, where M stands for macro and μ stands for micro. F_1 is F-measure, P is precision and R is recall.

Run	BLEU		Typeless F_1	LOCATION	PERSON	ORG	MISC
Baseline	26.14	μ	68.61	26.97	13.82	29.69	17.52
		M	69.82	71.20	55.83	60.57	60.31
Disc-Re	27.34	μ	69.50	27.31	14.25	29.99	17.85
		M	70.65	72.07	57.01	61.39	61.40

Table 6.4: Results of the Stanford NER system for each entity type on the German to English test set, where M stands for macro and μ stands for micro. F_1 is F-measure.

improved by the discriminative model are different from those involved around named entities.

The difference between the baseline and the reordering model runs for Bulgarian and Greek languages are insignificant. The main reason for this is due to lack of reordering during the translation from both languages. The best distortion parameter for Greek was 0 and for Bulgarian was 2, which explains the small difference between the two systems in terms of BLEU and *F*-measures.

The most substantial difference in named entity extraction quality is achieved in the case of translating from German to English. The main reason for this is due to the better quality of the translation with the help of the reordering model. Translating from German to English required the distortion limit of 9, and 19 with the reordering model. Therefore, the reordering model enables the decoder to perform longer-distance movements and resolve some of the problems of named entity extraction caused by the wrong context. It is also important to notice that the German-English test set is the only test set which is a mixture of sentences from two different domains namely, news commentary and Europarl.

Table 6.4 shows the breakdown of results for each entity type for the German to English test set. The improvement of NER occurs for all four types and neither of them benefit significantly more than the others.

6.6 Discussion

We investigated the feasibility of extracting named entities from machine translation output and the effect of reordering and improving it on the quality of named entity recognition.

The results showed that improving reordering for German to English translation can help both translation quality in terms of BLEU score and the quality of NER. On the other hand, the improved translation of Turkish to English did not lead to better NER. Therefore, not always better reordering and translation with respect to BLEU or similar metrics results in higher quality for a task-based metric such as NER.

Although Stanford's NER system has a higher quality than LingPipe, the differences between the results for different models of translation were consistent. This shows even though they use different classification models to label the input and Stanford's NER classifier uses more features, there are certain aspects of the translation output that affect named entity extraction, which are common for both systems.

We are aiming to extend this work to include the effect of other aspects of the translation process such as handling out-of-vocabulary words. In addition, we are going to to investigate other NLP tasks such as extracting grammatical relations with the same method.

Summary

We proposed a method to evaluate the effect of improving translation quality on a natural language processing task such as named entity recognition. The fact is, it is not easy to obtain parallel multi-lingual annotated data with named entities for evaluation. Therefore, we used the output of an automatic NE recogniser as the gold standard. In other words, our method evaluates the quality of a named entity extractor on a translated English sentence compared to the quality of the same extractor on the human generated English of the same sentence.

Two stat-of-the art tools on a variety of languages were tested and specifically examined to see the effect of reordering on the performance of NER. Even though higher quality translations with respect to automatic metrics were available for most of the languages, not in all cases better translation led to better named entity recognition. We provided the analysis for the role of reordering to explain this phenomenon.

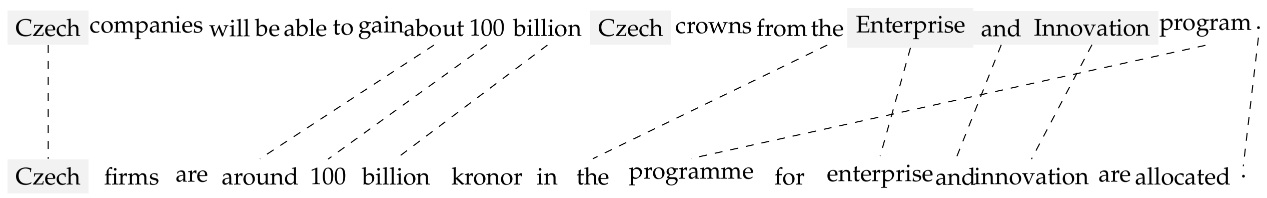

Czech companies will be able to gain about 100 billion Czech crowns from the Enterprise and Innovation program .
Czech firms are around 100 billion kronor in the programme for enterprise and innovation are allocated .

Figure 6.1: An example of word-aligned sentence pair. The above sentence is the reference sentence and the bottom is the translation. Extracted entities are highlighted.

CHAPTER 7

Cross-lingual Text Fragment Alignment using Divergence from Randomness

A notable portion of the information available on the Internet is given by documents which are obtainable from more than one source. For example, the same web page might be published on different mirror web sites, or the same piece of news could be reported, in slightly different versions, possibly in different languages. This phenomenon has several implications. In this chapter, we explore the use of statistical machine translation techniques to tackle the problems that arise from the cross-lingual nature of these documents. Similar to previous chapter that SMT methods were used to perform named entity recognition on foreign languages with tools tailored for English, in this chapter SMT methods are used along with similarity measures to perform text fragment alignment.

The remainder of this chapter is organised as follows: Section 7.2 provides a review of current research and methods in fields related to cross-lingual text alignment. Section 7.3 describes the alignment of text fragments algorithm and similarity measures to perform the sentence alignment. Construction of the test collection and experiments are reported in Section 7.4.

7.1 Why Text Alignment?

In the context of web search, data redundancy in the search results has already been shown to be an issue [Bernstein and Zobel, 2005]. For example, even if a document is considered to be relevant to an information need, when shown after a number of redundant documents, it does not provide the user any additional information. In other words, showing redundant documents does not benefit the user for the purpose of satisfying an information need.

A different point of view regards the versioning and the authorship of redundant documents. Given the dynamic nature of the Web, it is common to find different versions of the same document, e.g. pages which contain minor variations of another one. On the other side, plagiarising other authors becomes a very simple task. The task of identifying plagiarised documents, with a distinction between real plagiarism and mere topic similarity, is not trivial. Both plagiarism and versioning might affect a document as a whole, or just portions (e.g. sections, paragraphs, or more in general fragments) of it. An intelligent tool which helps in recognising duplicate text fragments could benefit editors and authors.

To tackle one aspect of these implications, this chapter investigates the possibility of aligning text fragments between documents written in two different languages. The main focus is identifying pairs of fragments with a strong content-based similarity. Figure 7.1 shows an example of aligning fragments of texts, which do not necessarily have the same length. Our approach, starts with measuring similarity at sentence level between the documents and then extract aligned fragments of texts based on the sentence similarities. The outcome will be a set of disjoint aligned fragments with the highest score based on the previously estimated sentence similarities.

The main component of our method is measuring the similarity between two text fragments. We have chosen models of information retrieval based on divergence from randomness to estimate the similarities and examine the best performing model in the context of cross-lingual text alignment. An advantage of models based on divergence consists in having multiple choices of randomness models, and hence the opportunity

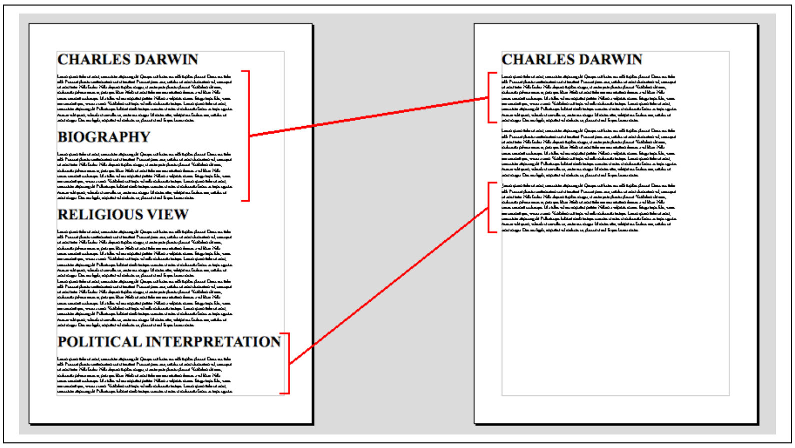


Figure 7.1: An example of aligned text fragments.

to evaluate many IR models for this task. In addition, these models are non-parametric and do not require parameter tuning and training data to perform well.

The information about the fragments of the documents produced by the alignment algorithm, can be used later for specific applications. Such applications include the possibility of automatically creating training data sets for machine translation or document summarisation, as well as automatically synchronising complex multi-lingual web sites (e.g. Wiki-based encyclopedias, or other user-driven sites). Previous work in this area has explored both novelty detection for improving search effectiveness, and the use of fingerprinting techniques for identifying redundant documents [Bernstein and Zobel, 2005], but mainly in a monolingual environment.

7.2 Related Work

This work lays on the overlap between the two areas of document summarisation and machine translation. Despite their differences in concepts and techniques, both summarisation and translation systems are mostly built on top of statistical methods, which require training data to acquire statistical patterns. [Daumé III and Marcu, 2004] propose an approach to automatically align documents to their respective summaries and extract transformation rules to shorten phrases to produce shorter and more informative summaries. Their algorithm is an extension to the standard HMM model and learns word-to-word and phrase-to-phrase alignment in an unsupervised manner.

In case of machine translation, availability of training data set is more crucial. Statistical machine translation, uses manually translated data in the forms of parallel sentences to learn translation patterns by statistical means. There has been extensive work focusing in finding parallel documents [Uszkoreit et al., 2010] and aligning sentences in fairly parallel corpora [Ma, 2006] and even non-parallel corpora [Munteanu and Marcu, 2005]. [Munteanu and Marcu, 2006] presents an approach to find sub-sentential segments from comparable corpora. Despite previous work, [Uszkoreit et al., 2010] propose a method that solely relies on textual content of the documents instead of metadata or document structure to find near-duplicate documents. All documents are automatically translated and n-gram features are extracted to construct a small set of candidate documents in a very large collection of documents. One-by-one comparison is performed using idf-weighted cosine similarity among the documents in the candidate set. They report that incorporating term frequency or other retrieval ranking functions degrade the performance compared to the mentioned similarity measure. Our approach is also based on textual content only, but the alignment is performed on fragments (see Section 7.3) rather than sentences or entire documents.

In cross-lingual plagiarism, the aim is finding fragments of text that have been plagiarised from the source document written in a different language. [Barrón-Cedeño et al., 2008] describe an statistical approach based on IBM model 1 [Brown et al., 1993] to retrieve the plagiarised fragment among a list of candidate fragments. The statistical approach is proposed to perform cross-lingual retrieval, bilingual classification and cross-lingual plagiarism and it focuses on the retrieval aspect of plagiarism. [Pouliquen et al., 2003] investigates the performance and effectiveness of different models of cross-lingual retrieval for the purpose of plagiarism detection. They compare retrieval models based on parallel and comparable corpora to models based on dictionaries and syntax of the languages involved. Similarly to [Barrón-Cedeño et al., 2008], IBM model 1 probabilities are used as translation probabilities in the statistical models and a length component is introduced to take into account the ration of length differences between the two languages.

Plagiarism detection has also been extensively investigated in a mono-lingual environment [Hoad and Zobel, 2003]. Similar work involves the identification of redundant [Bernstein and Zobel, 2005] and co-derivative [Bernstein and Zobel, 2004] documents, using fingerprinting techniques. Fingerprints are compact representations of text chunks. In these approaches, hash functions are used to calculate fingerprints of documents. Different documents are then identified as redundant, or as co-derivative, according to the fingerprint similarities. In our approach, the similarity is calculated on a fragment level, based on the content of the fragments.

7.3 Text Fragment Alignment

We define a text fragment as a list of continuous sentences in a document. Ideally, the content of a fragment is semantically coherent (i.e. it can be considered to be about a single topic). The aim of the proposed fragment alignment is to find fragment pairs in two documents, which are written in two different languages. Assume $\vec{d_e} =< s_{e_1}, s_{e_2}, \ldots, s_{e_n} >$ and $\vec{d_f} =< s_{f_1}, s_{f_2}, \ldots, s_{f_m} >$ are two documents in languages e and f, which contain n and m number of sentences respectively. We want to find a set of paired fragments that contains aligned text fragments that are related:

$$\{(\vec{\epsilon_i^{i'}}, \vec{\phi_j^{j'}}) | 1 \leq i \leq i' \leq n \wedge 1 \leq j \leq j' \leq m\} \tag{7.3.1}$$

where, $\vec{\epsilon_i^{i'}}$ represents a fragment that contains sentences i to i' from $\vec{d_e}$ and $\vec{\phi_j^{j'}}$ is a fragment that contains sentences j to j' from $\vec{d_f}$. Based on these definitions, fragments of a document can consist of different number of sentences and even relatively different number of sentences for each fragment in an aligned one. Since considering all the possible fragments in a document and aligning them with all the possible fragments in the other document is computationally very expensive, we restrict extracting the fragments by initial information about the alignment of sentences. The initial information is acquired by aligning sentences in the two documents and finding a few strong links between some of the sentences. A paired fragment can not contain a link to sentences outside the pair. This restriction significantly reduces the number of fragments that can be extracted.

Figure 7.1 sketches the text fragment alignment algorithm. The first step is to score all the sentence pairs and find a few links between the sentences. Next, all the fragments which are compatible with the links are extracted and sorted according to their scores. Finally, a set of non-overlap fragment pairs are selected as the output.

```
Input: $\vec{d_e}$ and $\vec{d_f}$ {$\vec{d_e}$ is English document, $\vec{d_f}$ is foreign document}
Input: similarity threshold $min\_score$
for all $s_{e_i}$ in $\vec{d_e}$ do
    for all $s_{f_j}$ in $\vec{d_f}$ do
        $score[i][j] \leftarrow$ estimate similarity between $s_{e_i}$ and $s_{f_j}$
        $link[i][j] \leftarrow (score[i][j] > min\_score)$
    end for
end for
$aligned \leftarrow$ extract fragment pairs compatible with $link$
$chosen \leftarrow \{\}$
for all $fragment$ in (sort $aligned$) do
    if $fragment$ overlaps with no member of $chosen$ then
        $chosen \leftarrow chosen \cup fragment$
    end if
end for
```

Algorithm 7.1: Text fragment alignment algorithm. *aligned* is the set of all aligned fragments and *chosen* is the final set of selected fragments.

7.3.1 Similarity Measures and Divergence from Randomness

A major step in finding aligned fragments of two documents is estimating similarity between sentences. As pointed out in the introduction, we have chosen a set of probabilistic models of information retrieval based on divergence from randomness [Amati and Van Rijsbergen, 2002]. A basic assumption of DFR[1] models is that non-informative words are randomly distributed in the collection. In DFR, a randomness model M is chosen to compute the probabilities and there are many ways to choose M, such as Bose-Einstein distribution or Inverse Document Frequency model. $Prob_1(tf)$ is defined as the probability of observing tf occurrences of a term in a randomly selected docu-

[1] Divergence from Randomness

ment according to M. Thus, if $Prob_1$ is relatively small for a term, then the term is an informative one. Another probability, $Prob_2$, is defined as the probability of occurrence of a term within a document with regard to a set of documents that contain the term.

The term weight, under the above definitions is the product of two factors: Firstly, information content of the term with respect to the whole collection, which is formulated as $Inf_1 = -\log_2 Prob_1$. Secondly, $Inf_2 = 1 - Prob_2$, information gain of the term with respect to its elite set, which is the set of documents that contain the term.

$$w = Inf_1 \times Inf_2 = (-\log_2 Prob_1) \times (1 - Prob_2) \tag{7.3.2}$$

Here, we are computing the similarity between two sentences in two different languages, s_e and s_f. Terms in s_f are translated based on a lexical translation model and converted to a bag-of-word with, s'_f, translation probabilities for each term. The lexical translation model is based on the IBM model 1 described in [Brown et al., 1993]. The similarity between two sentences s_e and s_f is calculated as follows:

$$\mathrm{sim}(s_e, s_f) = \mathrm{sim}(s_e, s'_f) = \sum_{t \in \{s_e \cap s'_f\} \wedge \tau \in s_f} w_M(t, s_e) \times p(t|\tau) \tag{7.3.3}$$

where, $w(t, s_e)$ is the weight if term t in sentence s_e according to similarity model M and $p(t|\tau)$ is the translation probability of translating τ to t. The collection for Equation 7.3.3 is $\vec{d_e}$, which is the document that contains s_e and all the collection statistics in the similarity measures are computed based on $\vec{d_e}$. Table 7.1 shows a list of all the models used in this work to estimate the sentence similarity between two documents.

7.3.2 Extraction of Fragments

After scoring all the sentence pairs, only those with similarity score higher than a certain threshold are aligned. Aligned fragments are extracted by an algorithm adopted from phrase-based statistical machine translation [Och et al., 1999]. Fragments in an extracted fragment pair are only aligned to each other and not to any fragment outside the fragment pair.

	Name	Description
1	TF-IDF	The $tf.idf$ weighting function, where tf is the total term frequency and idf is Spärck-Jones' formulation
2	$\mathrm{TF_k}$-IDF	Same as above but with the BM25 tf quantification $\frac{tf}{tf+k}$
3	$I(n)L2$	Model with Inverse document frequency, with Laplace after-effect and 2nd normalisation
4	$I(F)B2$	Model with Inverse of the term frequency, with Bernoulli after-effect and 2nd normalisation
5	$I(n_e)B2$	Model with Inverse of the expected document frequency, with Bernoulli after-effect and 2nd normalisation in base 2
6	$I(n_e)C2$	Model with Inverse of the expected document frequency, with Bernoulli after-effect and 2nd normalisation in base e
7	$BB2$	Limiting form of Bose-Einstein, with Bernoulli after-effect and 2nd normalisation
8	$PL2$	Poisson approximation of the binomial model, with Laplace after-effect and 2nd normalisation
9	BM25b	BM25 probabilistic model
10	OkapiBM25	Okapi formulation of BM25

Table 7.1: Similarity measures used to estimate the similarity between sentences. For detailed information on each model, please refer to [Amati and Van Rijsbergen, 2002].

Many of the extracted aligned fragments overlap and there are sentences which belong to more than one fragment. Therefore, we sort all the aligned fragments according to their similarity score and drop those with lower scores and overlap. The score of an aligned fragment is estimated by averaging the similarity scores of its sentence pairs computed before. The remaining aligned fragments are the result of the algorithm.

7.4 Experimental Study

7.4.1 Experimental Set-up

Since we did not have a manually annotated documents with aligned fragments, a pseudo-collection is constructed to perform the experiments. A collection of documents and their summaries in English and Italian is built by crawling the web-site of the Press releases of the European Union[2] and pseudo-documents are created by randomly concatenating documents and summaries to each other. For the English side, x documents are randomly chosen and concatenated to create a document with multiple topics. On the Italian side, y documents are randomly chosen, added to the set of x aligned summaries of the chosen documents and randomly concatenated. As a result, we have an English document consisting of x documents and an Italian document consisting of $x + y$ summaries, including the summaries of the English documents. The task is now defined to be aligning all the sentences of the summaries to their correct English document or to not-align those with no corresponding document. Table 7.2 shows statistics of the corpus. All the documents and summaries in the collection are processed by tokenisation, lower-casing and sentence splitting.

7.4.2 Document-Summary Association

As a basic task compared to finding aligned fragments of text, we examine the problem of associating documents to their summaries. Association is the process of finding two related structures in a collection of structures. In a collection of documents and

[2] Available at http://europa.eu/rapid

	English	Italian	Average
Mean Document Length (sentences)	34.66	35.29	34.96
Mean Summary Length (sentences)	5.09	4.87	4.98
Mean Compression Ratio (sentences)	14.68%	13.81%	14.26%
Mean Document Length (words)	794.85	874.73	834.79
Mean Summary Length (words)	106.08	118.74	112.43
Mean Compression Ratio (words)	13.35%	13.58%	13.47%
Number of document/summary pairs	192		

Table 7.2: English-Italian corpus statistics

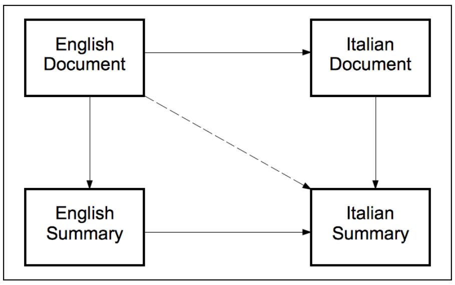


Figure 7.2: Cross-lingual Summarisation Pipelines: Two-Stage vs. One-Stage

summaries, the aim is to find the most related summary to each document. We assume that there is a one-to-one association between the summaries and the documents.

The association process can be performed in two ways. Firstly, a two-stage method which translates and summarises the document and computes the similarity between the summaries. Secondly, a one-stage cross-lingual association approach that directly calculates the similarity between the document and the summary in different languages. An illustration of English-to-Italian association is drawn in Figure 7.2, which shows the two ways that the association can be performed in. The one-stage approach estimates the similarity between the document and the summary according to equation 7.3.3, but instead of similarity between sentences, its the similarity between documents and summaries.

In the two-stage approach, the summarisation component relies on MEAD [Radev et al., 2004], which is an extractive summariser. The machine translation system used for translation form Italian to English is a phrase-based statistical MT system with translation model and language model as its main components. The full detail of the system is described in [Yahyaei and Monz, 2010b]. The training data for the SMT system is taken from the Europarl corpus [Koehn, 2005]. 1.6 million parallel sentences were used for building the translation model and 50 million sentences to train the English language model. For both approaches, lexical probabilities are estimated based on IBM model 1 and the parallel training data mentioned before.

The scores for the one-stage system, which associates English documents to Italian summaries, are shown in Table 7.3, where one can observe that the $OkapiBM25$ function is performing the best. The best scores for the two-stage method are P@1= 78.1% and MRR= 82.0 and the results of the two-stage approach are in all the cases substantially lower than the one-stage one.

Table 7.3: Results of document-to-summary association of the one-stage approach with different similarity measures.

Similarity	**P@1**	**MRR**
TF-IDF	88.6	92.1
TF_{k}-IDF	89.1	92.4
IDF	86.0	89.8
$BM25b$	89.6	93.0
$OkapiBM25$	**91.7**	**94.3**
$I(n)L2$	90.1	93.1
$I(F)B2$	81.8	86.9
$I(n_e)B2$	86.5	90.6
$I(n_e)C2$	86.0	89.9
$PL2$	90.1	93.2

In the two-stage approach approach, the summarisation and translation tasks lead to a loss of information which cannot be adequately captured by the association functions we have examined. After performing the association of English summaries and MEAD generated summaries from the documents, a basic similarity measure such as TF-IDF achieved a P@1 score of 98.0 and MRR of 99.2. This means that the translation component is the major source of precision loss in the two-stage method. The translation component translates each Italian sentence to exactly one English sentence. For translating each sentence, it selects the translation with highest score according to its model to produce a fluent English. The produced sentence only contains one possible translation for each word or phrase. On the other hand, the one-stage approach considers all the possible translations in the lexical model for each word, hence having a higher chance of finding a match between document words and summary words. The 91% success rate of the one-stage approach, shows it is possible to associate the majority of the summaries to their documents in this collection. The results of the text fragment alignment show the difficulty of finding the same summaries, while they are mixed with other summaries.

7.4.3 Text Fragment Alignment Evaluation

To find out the cross-lingual effect of the task, we performed the text fragment alignment algorithm on mono-lingual data as well as the cross-lingual data. For each word only the top 5 translations based on the their translation weights are picked. The threshold is set to the average score of the alignment links, therefore alignment links with score less than the average are discarded. For each similarity measure, the alignment algorithm is run $2,000$ times to select different variations of the documents and summaries.

The goal of text fragment alignment is to find the longest relevant fragments of text on each side, without including irrelevant sentences. Therefore, both recall and precision are important in evaluating the algorithm. F-measure combines the two, to give one single score to demonstrate the performance of the algorithm. To calculate the F-measure, each sentence on the e side is labelled true positive if it belongs to a fragment, which is fully or partially correctly aligned. The sentence is labelled false positive if it belongs to a fragment which is incorrectly aligned. It is a false positive instance, if it is not aligned and it should not have been. A false negative instance is an unaligned sentence, which should have been aligned. F-measure is calculated based on these labels for both sides, English to foreign and foreign to English.

Table 7.4 shows the results of both mono-lingual and cross-lingual text fragment alignment experiments. As expected, the results of the mono-lingual text fragment alignment are higher than the cross-lingual runs. In all settings and in both directions

	Mono-lingual				Cross-lingual			
	μF_1	μF_1	MF_1	MF_1	μF_1	μF_1	MF_1	MF_1
Similarity	**src2trg**	**trg2src**	**src2trg**	**trg2src**	**src2trg**	**trg2src**	**src2trg**	**trg2src**
TF-IDF	33.5	77.0	34.9	77.0	23.0	28.5	23.6	27.0
TF_k-IDF	35.2	76.4	35.4	76.3	22.4	28.7	21.5	26.6
$I(n)L2$	35.8	80.2	35.7	80.3	30.0	**32.9**	29.4	**31.8**
$BB2$	34.5	**88.1**	34.9	**88.0**	27.6	32.3	28.2	31.4
$I(F)B2$	35.0	81.9	35.2	81.9	27.4	31.6	27.9	30.4
$I(n_e)B2$	34.9	74.3	35.4	74.2	27.9	31.9	28.4	30.7
$I(n_e)C2$	**38.3**	71.7	**38.2**	71.5	29.0	31.4	28.7	30.3
$PL2$	35.8	79.1	35.5	79.0	29.6	32.5	28.8	31.2
$BM25b$	36.7	72.4	36.7	72.1	30.8	32.5	30.1	31.3
$OkapiBM25$	37.3	71.3	37.5	71.1	**31.5**	31.9	**31.0**	31.0

Table 7.4: The results of text fragment alignment, for mono-lingual and cross-lingual. For mono-lingual, source and target (src2trg and trg2src) are both English documents and summaries. In cross-lingual settings, source is English documents and target is Italian summaries.

(source to target and target to source), models based on DFR substantially outperform TF-IDF weighting methods. In both mono-lingual and cross-lingual runs $OkapiBM25$ performs consistently very well compared to others. It has been pointed out by [Amati and Van Rijsbergen, 2002] that BM25 formula can be derived from the model $I(n)L2$, which has the highest score in the target to source cross-lingual runs and it is very close to other BM25 scores. Substantial drop of F-measure score of the target to source direction of the cross-lingual runs compared to mono-lingual ones, shows that the summary to document alignment is more prone to translation than the other direction.

Two important components of all similarity methods used in these experiments, are document length (sentence length in this work) and average document length in the collection. These factors are considered to reduce the effect of variance in document length in text collections, however, in our experiments, documents are sentences and they tend to be very similar in terms of number of words. We investigated two other ways to estimate sentence length and used them instead of the default number of tokens. One is sum of the term frequency in the collection for each term in the sentence[3] and the other one, sum of their selectivity (one over IDF)[4]. Both methods produced different results for all the runs, however, they were most of the times slightly worse than the number of tokens, and in general the differences were negligible. Sum of the selectivity of the terms perform slightly better for TF-IDF similarity, but in all other cases was behind the number of tokens. We concluded that the DFR models perform well in the context of sentence similarity even though, there is a difference between sentence length variation and document length variation in large collections.

7.5 Discussion

We developed an algorithm to perform cross-lingual text fragment alignment and ran a series of experiments with different similarity measures based on models of divergence from randomness. The results show that term statistics based on divergence models are

[3] $\text{len}(s) = \sum_{t \in s} tf(t, \vec{d})$, where s is a sentence in $\vec{d}$.

[4] $\text{len}(s) = \sum_{t \in s} sf(t, \vec{d})^{-1}$, where s is a sentence in $\vec{d}$ and $sf(t, \vec{d})$ is the number of sentences in $\vec{d}$ that contain t.

consistently superior to TF-IDF schemes. Despite the fact that sentences tend to similar in terms of length, we discovered that other ways of estimating sentence length does not improve the quality of the alignment compared to the basic method of counting the number of the tokens. In addition, for the source to target alignment the cross-lingual scores were not substantially lower than the mono-lingual ones, which shows that the translation component performs well enough not to degrade the overall performance considerably.

Preliminary investigation of cross-lingual association of documents and their summaries showed that a one-stage direct computation of similarity using a probabilistic dictionary (lexical probabilities) significantly outperforms a method that translates and summaries the documents and estimates a mono-lingual similarity between the documents. Experiments on mono-lingual associating of generated summaries and manual summaries showed that the low performance of the two-stage method is mainly due to the selective nature of the translation component. One translation is chosen among a list of possible translations based on the context of the sentence and the rest of the candidates are discarded, therefore, the chance of a match between the words of the two documents are heavily degraded.

Although the scores of the basic similarity measures were lower than most of the models of DFR in the association task, the difference was not substantial. In other words, even the basic models of similarity performed well in finding the corresponding summary for a document in our experiments.

Summary

This chapter described an approach to automatically align fragments of texts of two documents in different languages. A text fragment is a list of continuous sentences and an aligned pair of fragments consists of two fragments in two documents, which are content-wise related. Cross-lingual similarity between fragments of texts is estimated based on models of divergence from randomness and a set of aligned fragments based on the similarity scores are selected to provide an alignment between sections of the two documents. Similarity measures based on divergence have shown strong performance in the context of cross-lingual fragment alignment in the performed experiments.

CHAPTER 8

Conclusions

8.1 Summary

This thesis has explored the reordering phenomenon in machine translation and specifically examined the reordering problem in the phrase-based statistical machine translation systems.

The thesis is structured in three sections. The first section overviews statistical machine translation and the main concepts of SMT training, decoding and evaluation. Then defines the reordering problem and its different types. In Chapter 2 a brief introduction to SMT is given and in Chapter 3, most of the current approaches to deal with the reordering directly or indirectly have been discussed.

In the second section, we proposed models to improve the SMT decoder capabilities in dealing with the reordering requirements of different languages. Chapter 4 proposes a technique to extend the decoder to perform chunking and translating in tandem. This extension gives the decoder the ability to move segments of words and make long distance jumps to capture structural reorderings such those required for SOV to SVO translation. In Chapter 5, a discriminative reordering model was built that benefits from several features such as words, bigrams, POS, chunking information, sentence punctuation and etc, to make reordering decisions. In addition, the model is used to enable the decoder to adjust the distortion limit and the reordering window according to the context of the phrase about to be translated. Experiments on Arabic-English, French-English and Turkish-English were carried out to prove the effectiveness of the approach.

The third section of the thesis deals with the evaluation of the SMT for specific applications. Particularly, this section investigates the effect of improving reordering on the performance of NLP tasks on the MT output. Chapter 6 proposes an approach to use the output of automatic tools on the reference translations as the gold standard and evaluate the output of the same tool on the translation. This method gives us an insight to the possibility of using the combination of translation and English-tailored tools to solve multi-lingual problems. Finally, in Chapter 7, a method is presented to align fragments of text of two documents that are in different languages and are not necessarily parallel. This chapter also investigates the effectiveness of divergence from randomness models in estimating the similarity between fragments of text in a cross-lingual setting.

8.2 Conclusions

We outline the conclusions of this work as answers to the research questions asked in the introduction chapter at the beginning of the thesis (see Section 1.1). Some of the questions are addressed together.

1. *How can one take advantage of the fact that words tend to move together when they are translated across languages?*

2. *Is chunking and grouping words together a helpful solution for long-distance reordering?* Dynamically chunking and at the same time translating segments of the sentence produces very different translations from the baseline and is an effective approach to deal with the problem of long-distance reordering, which is due to structural differences between the two languages, in German to English translation.

3. *How important and effective is language modelling in dealing with the reordering problem?*

Language models are not sufficient enough to deal with the reordering problem and the short context considered by n-gram language models are not able to capture the long distance dependencies of words. Despite the fact that language models are not enough in current state of the art SMT systems, they are one of the most important features of all the SMT decoders. The high weight given to the language model feature by the MERT optimisation algorithm indicates the importance of the language model. The n-gram language model, not only makes decision about the order of words, in a short distance based on the order of the model, it plays a crucial role in determining the word choices and morphologies.

4. *How can an important parameter be tuned to avoid using the same parameter for sentences with different structure?*

   The way defining the distortion window, which is called reordering constraint, is as important as the value of the distortion parameter. Experiments on various language pairs show the importance of the distortion parameter in the quality of translation and the need for properly adjusting it. The dynamic distortion method enables the decoder to make long distance jumps by compensating for them by avoiding unnecessary skips in other parts of the sentence. In addition, it eliminates the need for finding the optimal distortion parameter and reordering constraint by deciding about them before each phrase expansion.

5. *What kind of features in a reordering model help to relax the reordering constraints in phrase-based SMT without degrading the performance of the algorithm in terms of speed and quality?*

   Results of the experiments of discriminative reordering models with several features revealed a few points in these reordering models: Firstly, source side features are by far more effective and the target side features. One reason for this can be the fact that the target language model is already contributing evidence from the target side. Also, it is the target language is built partially and with uncertain and probably not perfect building blocks, so the features from the target side are not as reliable as the features from the source side. Secondly, surface form features, particularly bigram words are more important and useful than POS, chunk information and punctuation features. Part of speech and chunk information are entirely dependent on external tools that are not hundred percent precise, particularly for non-English languages. In addition, words as features are more specific than for example POS categories. Thus, even though more general categories are useful for generalisation, their precision declines compared to word features for reordering decisions.

6. *Does adjusting the distortion limit improve quality of the translation compared to manual tuning?*
   In our experiments there was no difference between the features of DISCRIM-REO and DYNAMIC-DL for determining the scores of different permutations, however the DYNAMIC-DL method performed better than the DISCRIM-REO model alone. Therefore, the improvements achieved is due to the change of the search space explored by the decoder because of the adjustments of the distortion limit. The results show that guiding the decoder during the search can also be effective in improving the quality of translation in addition to removing the need for tuning the distortion parameter.

7. *What is the effect of being cross-lingual on text fragment alignment and is the difference between the performance of the mono-lingual algorithm and cross-lingual algorithm substantial enough to rule out the full translation as a viable approach in performing fragment alignment?*
   Experiments on cross-lingual association of documents and their summaries showed that a one-stage direct computation of similarity using a probabilistic dictionary (lexical probabilities) significantly outperforms a method that translates and summaries the documents and estimates a mono-lingual similarity between the documents. Experiments on mono-lingual associating of generated summaries and manual summaries showed that the low performance of the two-stage method is mainly due to the selective nature of the translation component. One translation is chosen among a list of possible translations based on the context of the sentence and the rest of the candidates are discarded, therefore, the chance of a match between the words of the two documents are heavily

degraded.

Similarity measures based on the divergence from randomness models outperform the similarity measures based on TF-IDF for fragment alignment in the cross-lingual and mono-lingual contexts.

8. *What is the effect of improving reordering on different NLP tasks for different language pairs? and is improving the reordering going to improve the quality of these tasks for all language pairs?*

   Improving the reordering and translation quality in terms of automatic evaluation metrics such as BLEU can lead to better named entity recognition in some languages, but not in all cases. For the German-English pair, the improved reordering resulted in better NER, however even though there were substantial BLEU improvements on Turkish to English translation, the NER performance was degraded. This showed the difference in the nature of reordering for the two language pairs. For some other language pairs with very limited reordering requirements such as Greek-English and Bulgarian-English, the NER performance did not change significantly mainly because of similar outputs by the baseline and the reordering models.

   Although Stanford's NER system has a higher quality than LingPipe, the differences between the results for different models of translation were consistent. This shows even though they use different classification models to label the input and Stanford's NER classifier uses more features, there are certain aspects of the translation output that affect named entity extraction, which are common for both systems.

8.3 Limitations and Future Work

The following list outlines some of the limitations of this research and a few directions for addressing them in future investigations and also a few suggestions for future work.

- Although one of the advantages of the dynamic chunking approach (see Chapter 4) is its language independence and no need for syntax-based tools, one may argue that for languages with available tools it can be beneficial to take syntax into account. The method described in this work defines and performs the chunking solely on the word alignment information, however, the classifier can be modified to take syntactic features that are produced by a syntactic chunker into account. In addition, it was mentioned that the chunking method suffers from speed problems compared to the baseline. One way of dealing with this issue is to reorder the source sentence based on the most probable chunk boundaries and produce an $n-$bast list of reordered sentences as the input of the decoder. This method is close to other non-determinant source reordering approaches described in Section 3.2.2.

- In performing the evaluation of named entity recognition on machine translation output, we used a set of imperfect tools to estimate the evaluation metric. Firstly, an automatic named entity recogniser was used to extract reference named entities from the reference translations. Secondly, a word aligner that uses exact, stem and WordNet matches was employed to link the entities between the translation output and the reference translations. These two steps are imperfect and can generate harmful noises in the evaluation process. On one hand, using these automatic methods enables us to evaluate named entity recognition on several language pairs and also avoids huge efforts of manual annotation, on the other hand, the noises produced by these tools can be misleading in some of the sentences.

- In the text fragment alignment method, we used divergence from randomness models with some other retrieval models to find the fragments of text in the two documents that are related. Although the approach is effective in detecting fragments of text, it does not provide a solution to find the two documents to be processed. There are several applications that the documents are already known or can be known easily by simple methods. However, there are scenarios that we are looking for fragments of text in large collections. The document association method provided in Chapter 7 requires the similarity of all pairs of documents to be estimated, which makes the process infeasible for large collections. Therefore, another method must be employed to find the two candidate documents in a tractable manner and then use the fragment alignment method to extract the related fragments. Having said that the method is still applicable for a wide range of applications and small collections.

- In the experiments of this research, several language pairs were used. Most of the experiments were carried out on European languages, plus Arabic and Turkish. English was the target language in all the experiments. For empirical investigations that deal with reordering, a language pair that requires both short and long distance reordering is invaluable. On the other hand, if both languages are morphologically simple, reordering will account for most of the translation difficulties and complex morphology does not degrade translation performance. We argue that Persian, English language pair is a very good candidate for empirical investigation of reordering in machine translation. Persian has a relatively simple grammar and has Subject-Object-Verb word order. Normal sentences have Subject, Preposition, Object and Verb word order, however it can have a relatively free word order due to the fact that the parts of speech are generally unambiguous. In addition, prepositions and the accusative marker help disambiguate the case of a given noun phrase. Different word orders between Persian and English makes the effect of the reordering aspect of the translation very significant. Unfortunately, there is no Persian, English parallel corpus available. Therefore, we have to prepare the collection by using limited web resources. Apart from the limited resources, different encodings and different styles of writing make it more difficult to clean and tokenise the corpus. Persian morphology is mainly dominated by suffixes. Verbs contain tense, aspect and agree with subject in person and number. For example, different documents use space, zero-width space or no space to separate the suffixes from the verbs. This increases the vocabulary size and makes the sparsity problem more severe.

Bibliography

Yaser Al-Onaizan and Kevin Knight. Named entity translation: extended abstract. In *Proceedings of the Second International Conference on Human Language Technology Research (HLT 2002)*, HLT '02, pages 122–124, San Francisco, CA, USA, 2002. [107]

Yaser Al-Onaizan and Kishore Papineni. Distortion models 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 (ACL 2006)*, ACL-44, pages 529–536, Sydney, Australia, July 2006. Association for Computational Linguistics. [49, 50, 51, 59, 79, 83]

Alias-i. LingPipe 4.0.1. http://alias-i.com/lingpipe, 2010. URL `http://alias-i.com/lingpipe`. [114]

Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20:357–389, October 2002. ISSN 1046-8188. [125, 127, 133]

Necip Fazil Ayan, Bonnie J. Dorr, and Christof Monz. Neuralign: Combining word alignments using neural networks. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 65–72, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics. [27]

Bogdan Babych and Anthony Hartley. Comparative evaluation of automatic named entity recognition from machine translation output. In *Workshop on Named Entity Recognition for Natural Language Processing Applications. In Conjunction with the First*

International Joint Conference on Natural Language Processing (IJCNLP 2004), Sanya City, Hainan Island, China, March 2004. [107]

Bogdan Babych and Anthony Hartley. Sensitivity of automated MT evaluation metrics on higher quality MT output: BLEU vs task-based evaluation methods. In *Proceedings of the Sixth International Language Resources and Evaluation*, LREC '08, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). [107]

Ibrahim Badr, Rabih Zbib, and James R. Glass. Syntactic phrase reordering for English-to-Arabic statistical machine translation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, EACL '09, pages 86–93, Athens, Greece, April 2009. Association for Computational Linguistics. [47]

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *Readings in speech recognition*, pages 308–319, 1990. [33]

Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation And/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. [36, 110]

Alberto Barrón-Cedeño, Paolo Rosso, David Pinto, and Alfons Juan. On cross-lingual plagiarism analysis using a statistical model. In Benno Stein, Efstathios Stamatatos, and Moshe Koppel, editors, *Proceedings of the ECAI'08 PAN Workshop: Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 9–13, Patras, Greece, 2008. [123]

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, 1996. ISSN 0891-2017. [43, 60, 61, 62, 63, 85]

Yaniv Bernstein and Justin Zobel. A scalable system for identifying co-derivative documents. In *Proceedings of 11th International Conference on String Processing and Information Retrieval (SPIRE 2004)*, pages 55–67, Padova, Italy, October 2004. [124]

Yaniv Bernstein and Justin Zobel. Redundant documents and search effectiveness. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, pages 736–743, Bremen, Germany, November 2005. [121, 122, 124]

Alexandra Birch and Miles Osborne. LRscore for evaluating lexical and reordering quality in MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 327–332, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-71-8. [106]

Alexandra Birch, Miles Osborne, and Philipp Koehn. Predicting success in machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. [15, 56]

Alexandra Birch, Miles Osborne, and Phil Blunsom. Metrics for MT evaluation: evaluating reordering. *Machine Translation*, 24:15–26, March 2010. ISSN 0922-6567. [106]

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Comput. Linguist.*, 16(2):79–85, 1990. ISSN 0891-2017. [23, 25]

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June 1993. ISSN 0891-2017. [23, 25, 123, 126]

Ada Brunstein. Annotation guidelines for answer types, 2002. LDC2005T33, Linguistic Data Consortium, Philadelphia. [108]

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of BLEU in machine translation research. In *11st Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, Trento, Italy, April 2006. The Association for Computer Linguistics. [36, 105]

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, Uppsala, Sweden, July 2010. [112]

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. [36]

Bob Carpenter. Character language models for Chinese word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 169–172, Sydney, Australia, July 2006. Association for Computational Linguistics. [115]

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher Manning. Phrasal: a toolkit for statistical machine translation with facilities for extraction and incorporation of arbitrary model features. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, HLT-DEMO '10, pages 9–12, Los Angeles, California, June 2010. Association for Computational Linguistics. [53]

Boxing Chen, Mauro Cettolo, and Marcello Federico. Reordering rules for phrase-based statistical machine translation. In *Proceeding of the 3rd International Workshop on Spoken Language Translation*, IWSLT '06, pages 53–58, Kyoto, Japan, November 2006. [48]

Colin Cherry and Dekang Lin. A probability model to improve word alignment. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL 2003)*, pages 88–95, Morristown, NJ, USA, 2003. Association for Computational Linguistics. [27]

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pages 263–270, Morristown, NJ, USA, 2005. Association for Computational Linguistics. [24, 32, 52]

David Chiang. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, 2007. ISSN 0891-2017. [24, 27, 52, 53]

Michael Collins. Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 489–496, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. [109]

Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, ACL '05, pages 531–540, Ann Arbor, Michigan, USA, June 2005. Association for Computational Linguistics. [46, 47, 58, 60]

Josep Maria Crego and Jose B. Marino. Improving statistical MT by coupling reordering and decoding. *Machine Translation*, 20(3):199–215, September 2006. [48, 59]

Mathias Creutz and Krista Lagus. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. In *Publications in Computer and Information Science, Report A81*. Helsinki University of Technology, March 2005. [90, 94]

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, July 2002. [107]

John N. Darroch and Douglas Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, October 1972. [63]

Hal Daumé III and Daniel Marcu. A phrase-based HMM approach to document/abstract alignment. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 119–126, Barcelona, Spain, July 2004. [122]

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithms. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. [25]

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. What can syntax-based MT learn from phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods Innatural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 755–763, Prague, Czech Republic, June 2007. Association for Computational Linguistics. [45, 59]

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research (HLT 2002)*, HLT '02, pages 138–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. [36, 70]

Ljiljana Dolamic and Jacques Savoy. Retrieval effectiveness of machine translated queries. *Journal of the American Society for Information Science and Technology*, 61:2266–2273, November 2010. ISSN 1532-2882. [107]

Markus Dreyer, Keith Hall, and Sanjeev Khudanpur. Comparing reordering constraints for SMT using efficient BLEU oracle computation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 103–110, Rochester, New York, April 2007. Association for Computational Linguistics. [81]

Jakob Elming. *Syntactic Reordering in Statistical Machine Translation*. Phd thesis, Copenhagen Business School, Denmark, 2008. [47]

Jakob Elming and Nizar 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, Athens, Greece, March 2009. Association for Computational Linguistics. [47]

Marcello Federico and Mauro Cettolo. Efficient handling of n-gram language models for statistical machine translation. In *Proceedings of the Second Workshop on Statistical*

Machine Translation, pages 88–95, Prague, Czech Republic, June 2007. Association for Computational Linguistics. [38]

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, ACL '05, pages 363–370, Ann Arbor, Michigan, USA, 2005. Association for Computational Linguistics. [115]

George Foster, Roland Kuhn, and Howard Johnson. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 53–61, Sydney, Australia, July 2006. Association for Computational Linguistics. [32]

Heidi J. Fox. Phrasal cohesion and statistical machine translation. In *EMNLP '02: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, pages 304–3111, Morristown, NJ, USA, 2002. Association for Computational Linguistics. [44]

Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856, Honolulu, Hawaii, October 2008. [51]

Michel Galley and Christopher D. Manning. Accurate non-hierarchical phrase-based translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference (NAACL-2010)*, Los Angeles, CA, June 2010. [53]

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL-04)*, Boston, USA, May 2004. [24, 27, 44]

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*, pages 961–968, Morristown, NJ, USA, 2006. Association for Computational Linguistics. [44, 45]

Spence Green, Michel Galley, and Christopher D. Manning. Improved models of distortion cost for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 867–875, Los Angeles, California, June 2010. [80, 91]

Nizar Habash. Syntactic preprocessing for statistical machine translation. In *Proceedings of the Machine Translation Summit (MT-Summit)*, pages 215–222, Copenhagen, Denmark, September 2007. [47]

Timothy C. Hoad and Justin Zobel. Methods for identifying versioned and plagiarized documents. *J. Am. Soc. Inf. Sci. Technol.*, 54:203–215, February 2003. ISSN 1532-2882. [124]

Hieu Hoang and Philipp Koehn. Improving mid-range re-ordering using templates of factors. In *EACL*, pages 372–379, 2009. [85]

Fei Huang. *Multilingual named entity extraction and translation from text and speech*. PhD thesis, Carnegie Mellon University, Pittsburgh, USA, December 2005. [107]

Abraham Ittycheriah and Salim Roukos. A maximum entropy word aligner for Arabic-English machine translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 89–96, Vancouver, B.C., Canada, October 2005. Association for Computational Linguistics. [27]

Margaret King, Andrei Popescu-belis, and Eduard Hovy. FEMTI: creating and using a framework for MT evaluation. In *Proceedings of Machine Translation Summit IX*, pages 224–231, New Orleans, Louisiana, USA, September 2003. [106]

Kevin Knight. Decoding complexity in word-replacement translation models. *Comput. Linguist.*, 25(4):607–615, 1999. ISSN 0891-2017. [34, 55, 80]

Philipp Koehn. *Noun Phrase Translation*. PhD thesis, University of Southern California, December 2003. [107]

Philipp Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA 2004*, pages 115–124, Washington, District of Columbia, 2004. [44, 67]

Philipp Koehn. Europarl: A parallel corpus for statistical machine translations. In *MT Summit X*, pages 79–86, Phuket, Thailand, September 2005. [112, 130]

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, November 2009. [27, 44]

Philipp Koehn and Christof Monz, editors. *Proceedings on the Workshop on Statistical Machine Translation*. Association for Computational Linguistics, New York City, June 2006. [70]

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, NAACL '03, pages 48–54, Edmonton, Canada, May 2003. Association for Computational Linguistics. [23, 24, 27, 30, 44, 46, 58, 70, 90]

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *The Proceedings of the 2nd International Workshop on Spoken Language Translation*, IWSLT '05, Pittsburgh, PA, October 2005a. [56, 84]

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, David Talbot, and M. White. Edinburgh system description for the 2005 NIST MT evaluation. In *MT Eval Workshop 2005*, 2005b. [51, 79, 90]

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting* *of the Association for Computation Linguistics (ACL), Demonstration Session*, pages 177–180, Jun 2007. [44, 53, 58]

Shankar Kumar and William Byrne. Minimum Bayes-risk decoding for statistical machine translations. In Susan Dumais, Daniel Marcu, and Salim Roukos, editors, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL '04, pages 169–176, Boston, Massachusetts, USA, May 2004. Association for Computational Linguistics. [98]

Alon Lavie and Abhaya Agarwal. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June 2007. Association for Computational Linguistics. [36, 105]

Zhang Le. Maximum entropy modeling toolkit for Python and C++. `http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html`, 2004. URL `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`. [63, 65]

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 720–727, Prague, Czech Republic, June 2007. Association for Computational Linguistics. [48]

Zhifei Li, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March 2009. Association for Computational Linguistics. [53]

Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 104–111, New York City, USA, June 2006. Association for Computational Linguistics. [96]

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(3):503–528, 1989. ISSN 0025-5610. [63]

Adam Lopez. Statistical machine translation. *ACM Comput. Surv.*, 40:8:1–8:49, August 2008. ISSN 0360-0300. [24]

Adam Lopez. Translation as weighted deduction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 532–540, Athens, Greece, April 2009. Association for Computational Linguistics. [24, 81, 96]

Xiaoyi Ma. Champollion: A robust parallel text sentence aligner. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC)*, Genova, Italy, 2006. [123]

Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceeding of the 6th Conference on Natural Language Learning*, COLING '02, pages 1–7, Taipei, Taiwan, August 2002. Association for Computational Linguistics. [63]

Christopher D. Manning and Hinrich Schtze. *Foundations of Statistical Natural Language Processing*. The MIT Press, June 1999. ISBN 0262133601. [33]

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, EMNLP '02, pages 133–139, Philadelphia, PA, USA, July 2002. Association for Computational Linguistics. [30]

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of EMNLP-2006*, pages 44–52, Sydney, Australia, 2006. [27, 44, 45]

Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`, 2002. [63, 90, 99]

I. McCowan, D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner, and H. Bourlard. On the use of information retrieval measures for speech recognition evaluation. IDIAP-RR 73, IDIAP, Martigny, Switzerland, 2004. [35]

Robert C. Moore. Improving IBM word-alignment model 1. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 518, Morristown, NJ, USA, 2004. Association for Computational Linguistics. [27]

Robert C. Moore. A discriminative framework for bilingual word alignment. In *HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 81–88, Morristown, NJ, USA, 2005. Association for Computational Linguistics. [27]

Robert C. Moore and Chris Quirk. Faster beam-search decoding for phrasal statistical machine translation. In *Proceedings of MT Summit XI, European Association for Machine Translation*, Copenhagen, Denmark, September 2007. [96]

Dragos Stefan Munteanu and Daniel Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31:477–504, December 2005. ISSN 0891-2017. [123]

Dragos Stefan Munteanu and Daniel Marcu. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 81–88, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. [123]

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, January 2007. ISSN 0378-4169. [108]

Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification, 1999. [62]

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. [63, 65, 99]

Franz Josef Och. An efficient method for determining bilingual word classes. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, pages 71–76, Morristown, NJ, USA, 1999. Association for Computational Linguistics. [48]

Franz Josef Och. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA, 2003. Association for Computational Linguistics. [38, 71, 90, 98, 114]

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, 2003. ISSN 0891-2017. [26]

Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449, 2004. ISSN 0891-2017. [24, 26, 44]

Franz Josef Och, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, 1999. [23, 27, 126]

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA, 2001. Association for Computational Linguistics. [36, 70, 71, 90, 98, 105, 106, 114]

Michael Paul, Marcello Federico, and Sebastian Stücker. Overview of the IWSLT 2010 evaluation campaign. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the Seventh International Workshop on Spoken Language Translation*, IWSLT '10, pages 3–27, Paris, France, December 2010. [93, 100, 101, 112]

Bruno Pouliquen, Ralf Steinberger, and Camelia Ignat. Automatic identification of document translations in large multilingual document collections. In *In RANLP 2003 Ð Proceedings of the International Conference on ÔRecent Advances in Natural Language Processing*, pages 401–408, 2003. [123]

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. MEAD - a platform for multidocument multilingual text summarization. In *LREC 2004*, Lisbon, Portugal, May 2004. [129]

Kay Rottmann and Stephan Vogel. Word reordering in statistical machine translation with a POS-based distortion model. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, TMI '07, pages 171–180, Skvde, Sweden, September 2007. [48]

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002. [111]

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, AMTA '06, pages 223–231, Cambridge, Massachusetts, USA, August 2006. [36, 70]

Andreas Stolcke. SRILM–an extensible language modeling toolkit. In *Proceedings of 7th International Conference on Spoken Language Processing*, volume 2 of *ICSLP '02*, pages 901–904, Denver, USA, September 2002. [38, 96]

Christoph Tillmann. A unigram orientation model for statistical machine translation. In Susan Dumais, Daniel Marcu, and Salim Roukos, editors, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL-Short '04, pages 101–104, Boston, Massachusetts, USA, May 2004. Association for Computational Linguistics. [51, 59, 79, 84]

Christoph Tillmann and Tong Zhang. A block bigram prediction model for statistical machine translation. *ACM Trans. Speech Lang. Process.*, 4(3):6, 2007. ISSN 1550-4875. [51]

Antonio Toral and Andy Way. Automatic acquisition of named entities for rule-based machine translations. In *Proceedings of the Second International Workshop on Free/Open-*

Source Rule-Based Machine Translation, pages 37–44, Barcelona, Spain, January 2011. [107]

Roy Tromble and Jason Eisner. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August 2009. Association for Computational Linguistics. [52]

Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1101–1109, Beijing, China, August 2010. Association for Computational Linguistics. [123]

David Vilar, Daniel Stein, Yugi Zhang, Evgeny Matusov, Arne Mauser, Oliver Bender, Saab Mansour, and Hermann Ney. The RWTH machine translation system for IWSLT 2008. In *International Workshop on Spoken Language Translation 2008*, pages 108–115, Waikiki, Hawaii, October 2008. [49]

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, COLING '96, pages 836–841, Copenhagen, Denmark, 1996. Association for Computational Linguistics. [26]

Chao Wang, Michael Collins, and Philipp 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 (EMNLP-CoNLL)*, pages 737–745, Prague, Czech Republic, June 2007a. Association for Computational Linguistics. [46, 58]

Wei Wang, Kevin Knight, and Daniel Marcu. Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 746–754, Prague, Czech Republic, June 2007b. Association for Computational Linguistics. [45]

Warren Weaver. *Translation*. Technology Press of the Massachusetts Institute of Technology, Cambridge, Mass., and John Wiley & Sons, Inc., 1955. [23]

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403, 1997. ISSN 0891-2017. [45]

Fei Xia and Michael McCord. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 508–514, Geneva, Switzerland, August 2004. Association for Computational Linguistics. [45, 46, 47, 58, 59]

Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model 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*, ACL-44, pages 521–528, Sydney, Australia, 2006. Association for Computational Linguistics. [80]

Deyi Xiong, Min Zhang, Aiti Aw, and Haizhou Li. A linguistically annotated reordering model for BTG-based statistical machine translation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 149–152, Columbus, Ohio, June 2008. Association for Computational Linguistics. [45, 80]

Sirvan Yahyaei and Christof Monz. Decoding by dynamic chunking for statistical machine translation. In *MT Summit XII: Proceedings of the Twelfth Machine Translation Summit*, pages 160–167, Ontario, Canada, August 2009. [21]

Sirvan Yahyaei and Christof Monz. Dynamic distortion in a discriminative reordering model for statistical machine translation. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation*, IWSLT '10, pages 353–360, 2010a. [114]

Sirvan Yahyaei and Christof Monz. The QMUL system description for IWSLT 2010. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation*, IWSLT '10, pages 157–162, 2010b. [130]

Sirvan Yahyaei, Marco Bonzanini, and Thomas Roelleke. Cross-lingual text fragment alignment using divergence from randomness. In *Proceedings of the 18th edition of the International Symposium on String Processing and Information Retrieval (SPIRE)*, 2011. [21]

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530, Toulouse, France, July 2001. Association for Computational Linguistics. [24, 27, 44]

Omar F. Zaidan. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88, 2009. [98]

Richard Zens and Hermann Ney. Improvements in phrase-based statistical machine translation. In Susan Dumais, Daniel Marcu, and Salim Roukos, editors, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL '04, pages 257–264, Boston, Massachusetts, USA, May 2004. Association for Computational Linguistics. [32]

Richard Zens and Hermann Ney. Discriminative reordering models for statistical machine translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation*, pages 55–63, New York City, NY, June 2006. [52, 79, 85]

Richard Zens and Hermann Ney. Efficient phrase-table representation for machine translation with applications to online MT and speech translation. In *Human Language Technologies 2007: the Conference of the North American Chapter of the Association for Computat Ional Linguistics; Proceedings of the Main Conference*, pages 492–499, Rochester, New York, April 2007. Association for Computational Linguistics. [38]

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. Reordering constraints for phrase-based statistical machine translation. In *COLING '04: Proceedings* *of the 20th International Conference on Computational Linguistics*, page 205, Morristown, NJ, USA, 2004. Association for Computational Linguistics. [43, 55, 81]

Jiajun Zhang, Chengqing Zong, and Shoushan Li. Sentence type based reordering model for statistical machine translation. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 1089–1096, Morristown, NJ, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-44-6. [80]

Yuqi Zhang, Richard Zens, and Hermann Ney. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting*, Rochester, NY, April 2007a. [48, 58]

Yuqi Zhang, Richard Zens, and Hermann Ney. Improved chunk-level reordering for statistical machine translation. In *International Workshop on Spoken Language Translation*, Trento, Italy, October 2007b. [49]

Index