SYNTACTIC REORDERING
IN
STATISTICAL MACHINE TRANSLATION

Jakob Elming

Copenhagen
Business School
HANDELSHØJSKOLEN

PhD thesis submitted June 2008

til min lille otto

Summary in English

Reordering has been an important topic in statistical machine translation (SMT) as long as SMT has been around. State-of-the-art SMT systems such as Pharaoh (Koehn, 2004a) still employ a simplistic model of the reordering process to do non-local reordering. This model penalizes any reordering no matter the words. The reordering is only selected if it leads to a translation that looks like a much better sentence than the alternative.

Recent developments have, however, seen improvements in translation quality following from syntax-based reordering. One such development is the pre-translation approach that adjusts the source sentence to resemble target language word order prior to translation. This is done based on rules that are either manually created or automatically learned from word aligned parallel corpora.

We introduce **a novel approach to syntactic reordering**. This approach provides better exploitation of the information in the reordering rules and eliminates problematic biases of previous approaches. Although the approach is examined within a pre-translation reordering framework, it easily extends to other frameworks. Our approach significantly outperforms a state-of-the-art phrase-based SMT system and previous approaches to pre-translation reordering, including (Li et al., 2007; Zhang et al., 2007b; Crego & Mariño, 2007). This is consistent both for a very close language pair, English-Danish, and a very distant language pair, English-Arabic.

We also propose **automatic reordering rule learning based on a rich set of linguistic information**. As opposed to most previous approaches that extract a large set of rules, our approach produces a small set of predominantly general rules. These provide a good reflection of the main reordering issues of a given language pair. We examine the influence of several

parameters that may have influence on the quality of the rules learned.

Finally, we provide **a new approach for improving automatic word alignment**. This word alignment is used in the above task of automatically learning reordering rules. Our approach learns from hand aligned data how to combine several automatic word alignments to one superior word alignment. The automatic word alignments are created from the same data that has been preprocessed with different tokenization schemes. Thus utilizing the different strengths that different tokenization schemes exhibit in word alignment. We achieve a 38% error reduction for the automatic word alignment.

Resumé på dansk

Omrokering af en sætnings ordrækkefølge har været et vigtigt område inden for statistisk maskinoversættelse (SMT) lige så længe som SMT har eksisteret. Dagens standard i SMT-systemer som fx Pharaoh (Koehn, 2004a) bruger stadig en forenklet model af processen bag omrokering i ikke-lokale omrokeringer. Denne metode devaluerer enhver omrokering, uanset hvilke ord den omhandler. Den eneste måde en omrokeringen kan blive gennemført, er hvis den fører til en oversættelse, der ligner en meget bedre sætning end alternativet. Uden hensyn til kildesætningen.

Nyere forskning har dog påvist forbedringer i kvaliteten af oversættelse ved brug af syntaksbaseret omrokering. En af disse metoder er omrokering før oversættelse, der inden den egentlige oversættelse justerer kildesætningen så dens ordstilling kommer til at ligne målsprogets ordstilling. Denne justering er baseret på regler, som enten er manuelt udformede, eller som er lært automatisk ud fra ordaligneringer mellem parallelle korpora.

I afhandlingen introduceres **en ny fremgangsmåde til syntaktisk omrokering**. Denne fremgangsmåde tillader en bedre anvendelse af informationen i omrokeringsreglerne og eliminerer problematiske uligevægtigheder i tidligere metoder. Selvom metoden undersøges inden for rammerne af omrokering før oversættelse, kan den nemt overføres på andre tilgange. Vores tilgang overgår et moderne frasebaseret SMT-system samt tidligere tilgange til omrokering før oversættelse, deriblandt (Li et al., 2007; Zhang et al., 2007b; Crego & Mariño, 2007). Dette gælder både for et nært beslægtet sprogpar, engelsk-dansk, og et meget forskelligt sprogpar, engelsk-arabisk.

Vi foreslår desuden en tilgang til **automatisk læring af omrokeringsregler baseret på en omfattende samling lingvistiske træk**. I modsætning til de fleste tidligere tilgange, der udtrækker et stort antal regler, produc-

erer vores tilgang kun et mindre antal overvejende generelle regler. Disse leverer et godt billede af de primære forskelle i ordrækkefølge inden for et givet sprogpar. Vi undersøger desuden betydningen af flere parametre, der kan have indflydelse på kvaliteten af de lærte regler.

Til sidst introducerer vi **en ny metode til forbedring af automatisk ordalignering**. Denne ordalignering bruges i det føromtalte arbejde med automatisk læring af omrokeringsregler. Metoden lærer ud fra manuelt alligneret data at kombinere flere automatiske ordaligneringer til én overlegen ordalignering. De automatiske ordaligneringer udledes af det samme datasæt, der er blevet præprocesseret med forskellige grader af ordsegmentering. På den måde udnyttes de forskellige styrker som forskellige ordsegmenteringer besidder i forbindelse med ordalignering. Dette fører til en fejlreduktion på 38% for den automatiske ordalignering.

Acknowledgements

I owe a lot of this to my supervisor and business partner, Dan Hardt, for being the eternal optimist. Thank you for support, advice, and always keeping your door/mind open. I am deeply indebted to Nizar Habash, my connection to the wonderful world of machine translation and the wonderful city of New York. Thank you for great collaboration and being such a good host. Also thanks to my colleagues at Computational Linguistics for creating a great place to work despite the hardships. I also feel very privileged in having an incredible family, who is always there for me. This also goes for all my friends at IAAS, Reaktoren, and Næstvedgade. Finally, I am eternally grateful for having Line in my life. You make me happy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> The problem of MT is *only* one of quantity and capacity.
>
> (Bar-Hillel, 1965 [1955], p.183)

These somewhat surprising words from Bar-Hillel are part of a thought experiment. His idea is that since language has a fixed vocabulary and maximal sentence length, only a finite set of sentences can be produced, and these can be pre-translated into a sentence dictionary, if the financial and technological means are at hand.

In theory, his idea seems flawed. First of all, vocabularies contain open word classes that are constantly changing. Through the news, we are for example constantly bombarded with new proper nouns. Secondly, sentence length has no limit. Any sentence can be prolonged, for example with *'He said X'*.

In practice, it may be true that with a fixed vocabulary and sentence length, almost all sentences can be covered, but even then, the idea of translating every sentence is inconceivable. Against a background of all English sentences ever produced, we would still expect this thesis to contain completely novel ones. Perhaps some of the sentences discussing Danish and therefore containing Danish words. This is why we need *machine translation* (MT). It possesses the ability to translate previously unseen sentences based on generalization as opposed to Bar-Hillel's idea, which fits better under the predicate translation memory.

If we disregard the problems of his idea, the words of Bar-Hillel in fact capture a divergence in *statistical MT* (SMT) today. Where some believe that

abstract linguistic knowledge, i.e. knowledge beyond word forms and their order, is needed for SMT to advance, others believe it is mainly a question of the quantity of data available to learn from. At the NIST MT Workshop 2006 in Washington, Daniel Marcu of the University of Southern California's Information Sciences Institute expressed his belief in integrating syntactic knowledge into SMT, while Franz Joseph Och of Google argued that simplifying the system and scaling up the amount of learning data was the way to go.

It is our belief that for certain aspects of translation, abstract linguistic knowledge is not only desirable, it is also necessary. Linguistic knowledge can reduce the need for massive amounts of data by raising the level of generalization, and thereby providing a basis for more efficient data exploitation. This is especially desirable for language pairs where massive amounts of data is not available. Moreover, there are systematic aspects of translation requiring information that is simply not available at the surface.

One such aspect of translation is *reordering*, which most often operates at a syntactic level. Syntactic information is therefore of the utmost importance. This is illustrated by the verb second nature of Danish, where the subject appears after the verb, if another constituent has taken its sentence initial position. Under the circumstances the subject of an English sentence stays in front of the verb. When these conditions are met, an English-Danish MT system must therefore reorder the subject. Example (1) illustrates this point (examples are annotated with English gloss and translation).

(1) idag er vi vidner til ...
[today are we witnesses to ...]
'today we are witnessing ...'

If the MT system has no abstract linguistic knowledge, it can neither identify the subject, verb or conditions. In this example, the condition is represented by *'idag'*, the subject by *'we'*, and the verb by *'are'*. Most state-of-the-art SMT systems will handle this by word sequence information. For example, information that *'today we are'* should be translated as *'idag er vi'*, or that the sequence *'idag er vi'* is more likely than *'idag vi er'*.

The condition can however vary from a word to an entire sentence, and

the subject can be any noun phrase from a word to a complex phrase containing a relative clause. Example (2) is comparable to example (1), where the subject moves after the verb, but here the condition is a sentence, and the subject is a complex noun phrase with a modifying prepositional phrase.

(2) som de ved, rådede retsudvalget mig ...
[as you know, advised the committee on legal affairs me ...]
'as you know, the committee on legal affairs advised me ...'

This example is very unlikely to be handled well based on word sequence information alone, since the sequence of words needed to cover both the condition and the reordering elements — the subject and the verb — is very long and unlikely to have been learned by the SMT system. A simple rule stating that the subject should reorder with the verb, if it is preceded by another constituent, would on the other hand be able to handle the reordering in both example (1) and (2).

State-of-the-art in SMT is at the moment *phrase-based SMT* (PSMT). Here the basic unit of translation is not a word but a word sequence. In connection with example (1), we illustrated that the unit *'today we are → idag er vi'* might be used to translate the example. As this example reveals, the sequences in PSMT need not be syntactically defined units. In our view, it is a strength of PSMT that it does not limit itself to syntactically defined units. We believe it is important to retain these syntax-independent strengths of PSMT when integrating syntax-based reordering.

One approach that retains these strengths while doing syntax-based reordering, is *pre-translation reordering*. Here the source sentence is reordered by syntactic rules prior to translation in order to obtain the word order of the target language. The core PSMT system is therefore not modified as such. For these reasons, we choose to test our approach within this framework.

A major deciding point in connection with pre-translation reordering is whether the reordering is imposed (*deterministic*) or merely suggested (*non-deterministic*). The deterministic approach reorders the source sentence and presents the PSMT system with this, while the non-deterministic approach presents the PSMT system with both the reordered and original sentence.

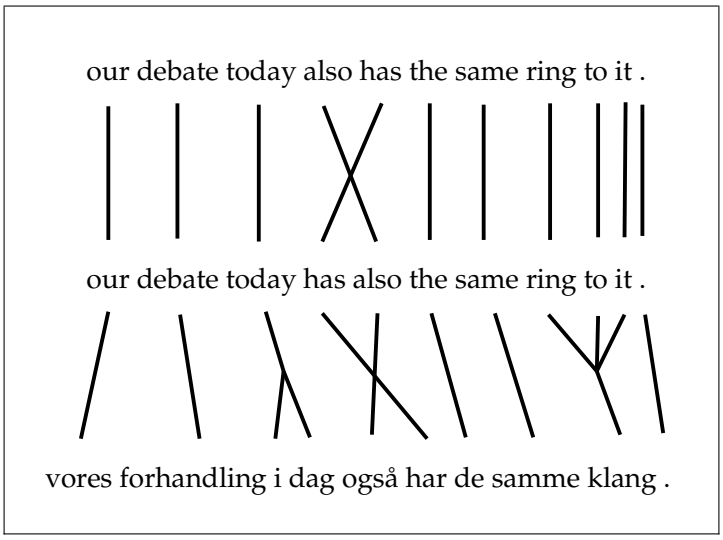


Figure 1.1: Example illustrating the problems with pre-translation reordering.

These suggestions are possibly *weighted* with different probabilities based on the rules.

We believe that the irrevocable decisions made by the deterministic approach are undesirable in a PSMT framework. Integrating the reordering decisions in the PSMT system instead yields a unified approach, where probabilistic reordering suggestions are evaluated together with all of the system's other parameters.

A major problem with pre-translation reordering approaches is that the information from the reordering rule is not carried over to the target language side. As a consequence, it has no influence on the decisions made by the system in the actual translation. Figure 1.1 illustrates this problem. The translation was produced by an actual pre-translation reordering system. The first sentence is the original English sentence, the second sentence is the pre-translation reordered source sentence, and the third sentence is the system output. Here a rule has moved the adverb *'also'* after the finite verb *'has'*, which yields the correct word order for Danish. Having accomplished the reordering, the rule rewards the translation. Following this, the system however reverts the reordering, but the translation retains the reward it was assigned for following the rule, even though this no longer is the case.

We address the problems of previous approaches to pre-translation reordering in PSMT by introducing a bisect approach. First, reordering rules generate multiple, unweighted source reorderings. Secondly, the rules are used to score the word orders of the translation hypotheses relative to the source sentence word order. Thereby, the word order of the translation is evaluated directly, and not through the word order of the reordered source sentence.

For the experiments, we use reordering rules that are automatically learned from word alignments based on multiple levels of linguistic knowledge. The experiments are carried out on a close language pair, English-Danish, and a distant language pair, English-Arabic. Here, our approach significantly outperforms a state-of-the-art phrase-based SMT system and previous approaches to pre-translation reordering measured by both human and automatic evaluation.

1.1 Contributions

We introduce **a novel approach to syntactic reordering**. This approach provides better exploitation of the information in the reordering rules and eliminates problematic biases of previous approaches. Although the approach is examined within a pre-translation reordering framework, it easily extends to other frameworks. Our approach significantly outperforms a state-of-the-art phrase-based SMT system and previous approaches to pre-translation reordering, including (Li et al., 2007; Zhang et al., 2007b; Crego & Mariño, 2007). This is consistent both for a very close language pair, English-Danish, and a very distant language pair, English-Arabic.

We also propose **automatic reordering rule learning based on a rich set of linguistic information**. As opposed to most previous approaches that extract a large set of rules, our approach produces a small set of predominantly general rules. These provide a good reflection of the main reordering issues of a given language pair. We examine the influence of several parameters in connection with rule learning

Finally, we provide **a new approach for improving automatic word alignment**. This word alignment is used in the above task of automatically learning reordering rules. Our approach learns from hand aligned

data how to combine several automatic word alignments to one superior word alignment. The automatic word alignments are created from the same data that has been preprocessed with different tokenization schemes. Thus utilizing the different strengths that these different tokenization schemes exhibit in word alignment. We achieve a 38% error reduction for the automatic word alignment.

1.2 Thesis outline

The thesis has the following structure:

- Chapter 2 provides the background for the work we present. It starts with a wide scope by looking at the field of MT, and then narrows in on SMT, ending up with PSMT. We also touch upon the role of linguistics in PSMT and the evaluation of MT.

- Chapter 3 deals with reordering in translation. We provide analysis of the language pairs involved in our experimentation, and discuss problems with reordering in PSMT and pre-translation reordering.

- Chapter 4 introduces a novel approach for improving the quality of automatic word alignment by exploiting multiple tokenization schemes. Word alignment plays an important role in learning reordering rules. The alignments produced here are used to learn reordering rules in chapter 5.

- Chapter 5 describes a novel approach for learning reordering rules based on a high level of linguistic knowledge. We experiment with learning from different data and word alignments.

- Chapter 6 presents a novel approach for doing syntactically motivated reordering in PSMT. The approach avoids the problems of previous approaches described in chapter 3.

- Chapter 7 concludes on the work done in the thesis and discusses perspectives for future work.

1.3 Publications

The work presented in chapter 4 is also described in *"Combination of statistical word alignments based on multiple preprocessing schemes"* presented at the North American Chapter of the Association for Computational Linguistics (Elming & Habash, 2007), which is joint work with Nizar Habash. This work is also extended in a chapter of the volume *"Learning Machine Translation"* (Elming et al., to appear) in participation with Josep M. Crego.

Chapter 5 and 6 expands on *"Syntactic reordering integrated with phrase-based SMT"* accepted for presentation at the Association for Computational Linguistics workshop on Syntax and Structure in Statistical Translation 2008 (Elming, 2008), and an elaborated version under the same title accepted for presentation at the International Conference on Computational Linguistics 2008 (Elming, accepted for publication).

Chapter 2

Background

This chapter provides the basic background for the work done in the thesis. The structure of the chapter is top-down. We start out by looking at what MT is, and how it is approached. Then we focus on SMT, giving a more detailed account of word-based SMT and PSMT. Following this, we look at how abstract linguistic knowledge can be integrated in PSMT. Finally, we will touch upon the difficult question of how to evaluate the performance of an MT system, both manually and automatically.

2.1 Machine translation

Machine translation deals with the automation of translation from one language to another. Given that language has both form and content, translation can be viewed as the representation of the same content in the form of different languages. Machine translation rests on a basic assumption, namely that different languages are able to express the same content. If we add a basic assumption of linguistics; that the languages have a systematic way of expressing the content, then we can also assume a systematic relationship between the formal expression of different languages.

These assumptions are not controversial. If people are able to learn language and translate, then there must be some form of systematicity within these processes. Given the presence of the appropriate knowledge about the languages and the relationship between them, a computer should therefore be able to perform the translation. However, the information needed

for translation quite possibly involves all levels of linguistics from word to world knowledge. This makes translation an AI-complete problem to solve. A computer needs to be as intelligent as a human to translate at the level of a human.

Despite these overwhelming challenges, the idea of using a computer for the task of translation has been around almost as long as the computer itself. It was probably the first natural language task proposed to be handled by a computer, and this at a time where few saw the potential of a computer outside number handling.

The ideas emerged in the wake of the immense success of code breaking in the Second World War. In a 1949 memorandum, an American mathematician, Warren Weaver, poses the idea of viewing translation as a cryptography problem.

> [I]t is very tempting to say that a book written in Chinese is simply a book written in English which was coded into the "Chinese code." If we have useful methods for solving almost any cryptographic problem, may it not be that with proper interpretation we already have useful methods for translation?
>
> (Weaver, 2003 [1949], p.16)

One problem with this proposed approach is that the task of cryptography does not compare directly to translation. Cryptography operates within a much more controlled space where strict operations directly linking the two expression sides are the actual basis for the encrypted language. That is, one side is not a natural language. The encrypted language is defined by and does not exist without these operations. Natural languages, on the other hand, undergo constant change, most often independent of each other, and they do not possess this direct link between the expression sides. The analogy is nevertheless appealing, and, as we shall see below, it has played an important role in the development of the SMT approaches employed today.

Approaches to machine translation can be divided into the following categories: *rule-based*, *example-based*, and *statistical*. In the following, we have chosen to group the description of these using an essential distinction within computational linguistics; whether the approach is based on

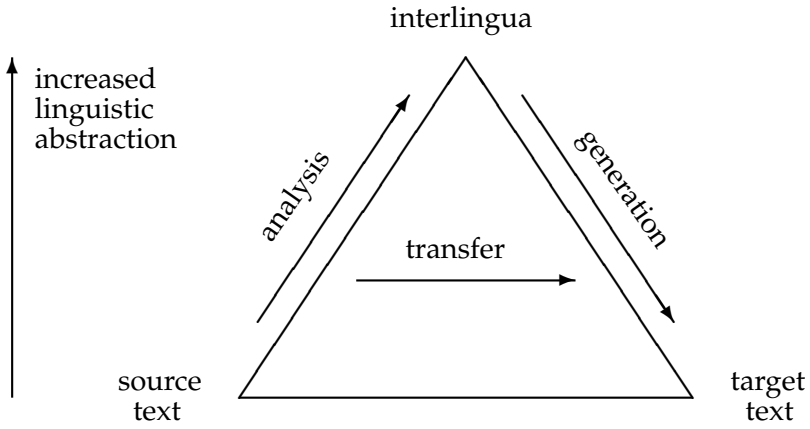


Figure 2.1: The Vauquois pyramid (based on Vauquois, 2003, Fig. 28.1).

the theoretical knowledge of linguistics or the information inherent in large corpora of text. In reality, the distinctions have become blurry, and are becoming even more so. Nevertheless, we feel this division is important to machine translation and provides the best basis for a description of the field.

2.1.1 Theory-based machine translation

For many years, the approach to machine translation was not based on the statistics of cryptography as proposed by Weaver, but on the introspection of linguists.

Also known as *rule-based machine translation*, this class of approaches is characterized by using rules based on several levels of linguistic abstraction. The process can be illustrated by the Vauquois pyramid in figure 2.1. First, the source text is subject to a linguistic analysis. This analysis yields an intermediate representation of the text. Then, based on this representation, a translation is generated using linguistic rules. These rule are manually crafted by experts with insight in both source and target language.

The intermediate representations can be of varying levels of linguistic abstraction. In the most abstract form, the intermediate level is an *interlingua*. That is, a pure representation of the text content that is common to all languages and thereby fully independent of the form of any single

language. This can be illustrated in figure 2.1 by going to the highest point of abstraction in the pyramid before generating the translation. This approach agrees with many people's intuitive notion of translation. When translating a sentence, you need to get the underlying idea, before you can proceed to produce the translation. The generality of the approach is also appealing. In theory, it should be possible to translate to every language from the same interlingua representation with the right rules.

In reality, it has so far not been possible to create this completely language-neutral representation (Hutchins & Somers, 1992, pp.123). This has lead most rule-based system to rely on linguistically less abstract intermediate representations. The lower abstraction level means that the representation is no longer language-independent. This is resolved by adding an additional step in translation that transfers the source representation to a target representation. These systems are called *transfer-based*. In figure 2.1, this is illustrated by cutting across to the generation side before reaching the top of the pyramid. The shape of the pyramid is supposed to reflect that the higher the level of linguistic abstraction, the simpler the transfer (shorter distance to the generation side).

Rule-based MT relies on the ability of experts to give an adequate formal description of the translation process. To a large extent, this not only entails a full internal description of the two languages, but also a description of how they correlate. The complexity of this task makes it seem an unreachable goal to achieve for any team of experts.

An advantage of the approach is that the developer has more control over the translations than is the case with corpus-based approaches. This level of control makes it possible for the developer to perform very focused modifications of the system's performance, and it provides the system with a high degree of consistency in its translations. As a consequence, the system has the advantage of high term consistency and predictable error correction (Elming, 2006). On the other hand, the high level of consistency restricts the system when doing less rigid task like ambiguity resolution. In addition, the creation of such a system is very time consuming, and as the system grows, the implications of a small modification may become unpredictable. Improving the translation of one phenomenon may hurt the translation of another.

2.1.2 Corpus-based machine translation

In the late 1980s, corpus-based approaches to machine translation started to emerge. Instead of solely relying on the knowledge of a small group of experts, researchers turned to the enormous amounts of information present in parallel corpora of texts that had already been translated.

In fact, the aforementioned theory-based approaches often used corpora to e.g. determine vocabulary or merely investigate language. What was new about the corpus-based approaches emerging in the late 1980s, was that they made use of the corpora in a much more direct way in translation. By piecing together instances of translation as encountered in the parallel corpus, the translation systems became a reflection of the language and translations they were based on.

Although the framework was introduced in 1984 under the name "translation by analogy" (Nagao, 1984), *example-based* MT (EBMT) was not attempted until 1990 (Sumita et al., 1990). This approach demonstrated the advantage of basing the translation of highly lexicalized, local phenomena on a parallel corpus.

Distinguishing EBMT from other approaches to MT is not a simple task. Somers (2003, p. 45) states that it is characterized by using example translations as its main knowledge source at runtime. This description is, however, disputed by Turcato & Popovich (2003, p. 64) stating that the origin of knowledge is unimportant, what is interesting is the knowledge that is used. At the end of the day, they only regard the original proposal (Nagao, 1984) as true EBMT, where the original, unprocessed examples are used at run-time. That is, the system has a direct link to the corpus at runtime. The distinction is very subtle, and we will take the less strict approach by leaning on Somers description.

In the extreme case, where the exact input sentence has been encountered previously, the EBMT system functions like a *translation memory*. A translation memory is a computer aided translation tool that is able to reuse previous translations. If the sentence or a similar sentence has been translated previously, the previous translation is returned. As opposed to a translation memory, the EBMT system is MT proper, in that it can translate novel sentences, and not just reproduce previous sentence translations.

In analogy to RBMT, EBMT translates in three steps; *matching*, *alignment*,

and *recombination* (Somers, 2003, pp. 7). 1) In matching, the system looks in its database of previous examples and finds the pieces of text that together give the best coverage of the input sentence. This matching is done using various heuristics from exact character match to matches using higher linguistic knowledge to calculate the similarity of words or identify generalized templates. 2) The alignment step is then used to identify which target words these matching strings correspond to. This identification can be done both using existing bilingual dictionaries or automatically deduced from the parallel data. 3) Finally, these correspondences are recombined and the rejoined sentences are judged using either heuristic or statistical information.

In example (3), Sato & Nagao (1990) illustrate the intuition behind EBMT by showing how it is possible to translate a previously unseen sentence based on translation examples. Sentence (3a) is translated by matching its parts in the translation examples (3b) and (3c). Via an alignment, the Japanese correspondences are located, and these are recombined into the translation (3d).

(3) a. He buys a book on international politics.
b. [He buys]$_b$ a notebook.
[Kare ha]$_b$ nouto [wo kau]$_b$.
c. I read [a book on international politics]$_c$.
Watashi ha [kokusaiseiji nitsuite kakareta hon]$_c$ wo yomu.
d. [Kare ha]$_b$ [kokusaiseiji nitsuite kakareta hon]$_c$ [wo kau]$_b$.

As can be seem from this description, EBMT is not a fully corpus-based approach. Especially, we quite possibly find theory-based components within matching and recombination. In this sense, EBMT can be positioned in between RBMT and SMT, which also causes the definition problems.

At about the same time that the first EBMT system was developed, the first SMT system was developed by a research team at IBM. This approach was based on the ideas proposed by Warren Weaver some 40 years earlier.

SMT has much in common with EBMT, but information is extracted by statistical means and annotated with probabilities. For a long time, SMT systems have been completely without linguistic knowledge (with the exception of the basic idea of a word token and a sentence), and in this pure

form, SMT stands as the prototype of a corpus-based MT approach. Within recent times, more statistical approaches are, however, beginning to integrate additional knowledge sources of higher linguistic abstraction.

The basic architecture of an SMT system also differs from that of an EBMT system. Generally, it will consist of at least a statistical translation model and a statistical target language model, and translation is done by letting a search algorithm, the *decoder*, examine a high number of possible translation, and the one yielding the highest probability based on all knowledge sources is considered the best translation. In the next section, we will proceed with a more detailed description of SMT.

2.2 Statistical machine translation

Some 40 years after Warren Weaver suggested using cryptographic and information theoretic methods for translation, the spirit of his ideas were taken up by a team of scientists at IBM. Stating that the main obstacles had been the power of computers and availability of electronic texts, they felt the time was ripe to pursue statistical machine translation (Brown et al., 1990).

2.2.1 Word-based SMT: The IBM models

Statistical machine translation is based on the idea that every target sentence is a possible translation of every source sentence. A reasonable assumption is that the probability of a given target sentence being a good choice for translation relies heavily on which source sentence is under consideration for translation. It is therefore possible to condition the probability on the source sentence, yielding the *posterior* probability of the target sentence given the source sentence $P(t|s)$[1].

This model of translation correlates with an intuitive view of translation; given a source sentence that we want to translate, which is the best target

[1] This description is based on (Brown et al., 1993). As opposed to their language specific notation (e stands for the English target string, and f stands for the French source string), we will use notation s for *source* string (i.e. the string to be translate) and t for *target* (i.e. the translation we looking for). We realize that this conflicts with other notation in their description, but we feel it brings more clarity in relation to the rest of the thesis.

sentence to choose as translation. That is, the target sentence that leads to the highest value for $P(t|s)$.

Following Bayes' theorem, the dependence between the sentences can be reversed:

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

Since the prior probability of the source sentence $P(s)$ is independent of, which translation is chosen, the denominator is constant. The target sentence that yields the highest value for the numerator, will therefore also yield the highest value for the entire equation. This leads to the equation Brown et al. (1993) calls *the fundamental equation of statistical machine translation*, where $\hat{t}$ presents the most probable target string:

$$\hat{t} = \underset{t}{\operatorname{argmax}} Pr(t)Pr(s|t) \tag{2.2}$$

As the name reveals, this equation is the backbone of the IBM models. It contains the story behind the generative models. A story that brings us back to Warren Weaver's idea. It tells how a source string s is generated. First, the target string t is generated, and then it is turned into the source string s. As proposed by Warren Weaver, we should therefore be able to retrieve the "original" target string t from the source string s. This approach is also know as the *noisy channel model*, since the target string is thought to have passed through a channel, where it has been corrupted by noise yielding a new string, the source string.

Equation (2.2) also unveils the basic components of SMT systems as mentioned in section 2.1.2; *the language model*, *the translation model*, and *the decoder*. $P(t)$ is the language model: a model of the target language. The basic function of this component is to ensure that the system produces acceptable target sentences. $P(s|t)$ is the translation model: a model of the relation between the two languages. This component ensures that suitable target words are used in the translation given the words of the source sentence. Finally, the maximization represents the decoder: the search problem of finding the target sentence that produces the highest probability given the language model and the translation model.

Instead of modelling the translation process directly as $P(t|s)$, Brown et

al. (1993) have distributed the influence to the two knowledge sources $P(t)$ and $P(s|t)$. The main reason for this choice is that their translation model is not strong enough to carry the entire translation. Two major weaknesses are the ordering of target words and relations between the target words. This weakness is to a large extent remedied by the language model, but in the process, the link to the information of the source sentence is weakened. That is, a lot of the votes when it comes to choosing the words and their order, are not directly based on the words and order of the source sentence.

In the following, we will describe the language model and the translation model as proposed by Brown et al (1993). We will, however, not go into the decoding problem here, since it is more a problem of efficiency than one of modelling the translation process.

Language model

The language model used for the IBM models is an n-gram model, which has proven to be a very strong source of information across natural language processing (see e.g. Jurafsky & Martin, 2006, ch.6). The model is very simple. It builds on the Markov assumption that the probability of a given word appearing in a sentence only depends on the previous n-1 words. Even though these decisions are very local, the model provides a good evaluation of the entire sentence, since the overlapping n-grams make up a chain of events. If n is set to three, the model is called a trigram model. Evaluating the sentence *'John loves Mary.'* with a trigram would be broken down in the following chain of probabilities (excluding artificial sentence boundary markers):

$$P(John) \times P(loves|John) \times P(Mary|John\ loves) \times P(.|loves\ Mary) \quad (2.3)$$

The model is learned by examining a large amount of monolingual text. This is done fairly simple by maximum likelihood estimation (see e.g. Manning & Schütze, 1999, sec.6.2.1). As an example, the probability of *P(Mary|John loves)* can be estimated by seeing how often the sequence *'John loves'* is followed by *'Mary'*. A better result is achieved by assuming that the training data perhaps does not reflect the language perfectly. The data can be smoothed out for example by discriminating infrequent n-grams. For a

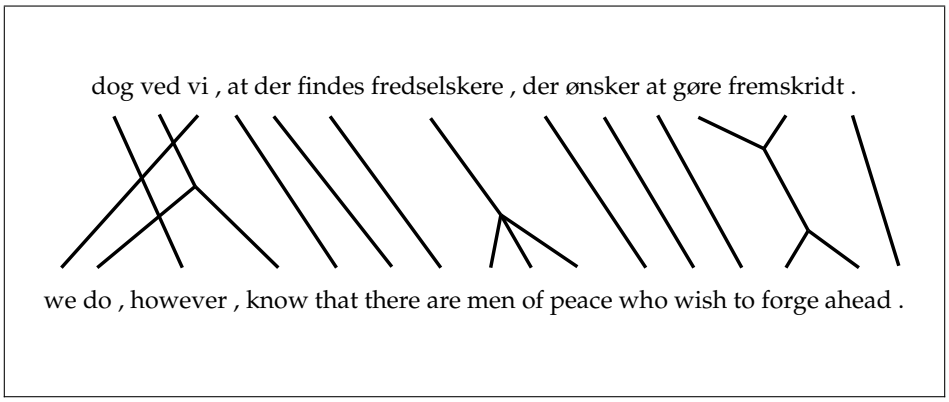


Figure 2.2: Example word alignment from a from the Copenhagen Danish-English Dependency Treebank (Buch-Kromann et al., 2007).

survey of different approaches to smoothing, see (Chen & Goodman, 1998).

Translation model

The IBM models only differ in their translation models. Since the translation model is modelled as the reverse process due to the Bayesian inversion, the following description of the translation model will appear as a description of how a target sentence is translated into the source language.

Brown et al. (1993) recast the problem of modelling translation through the problem of determining all possible *word alignments* a between two sentences:

$$P(s|t) = \sum_{a} P(s, a|t) \tag{2.4}$$

A word alignment is a representation of word level correlations between a source sentence and a corresponding target sentence, as exemplified by figure 2.2. Here we will use the term *(word) alignment* to mean the entire set of correlations between two word sequences, while a single correlation will be called a *link*. In figure 2.2, the words are linked to the words in the parallel sentence that they correspond best with. This may lead to anything from many words linking to many words, as is the case with the somewhat idiomatic construction *'forge ahead'*, to words linking to nothing. In this illustration, commas are for example not linked to anything, since they have

different functions in the two sentences[2].

The 5 IBM models developed to calculate the probability in equation (2.4) can be separated in two categories based on the generative story of how translation is performed. In the following, we will describe the IBM models based on this division. The first story makes the basis for the first two models, and the last three models are described though the second story.

IBM model 1 and 2. The first story is expressed in equation (2.5):

$$P(s, a|t) = P(m|t) \prod_{j=1}^{m} P(a_j|a_1^{j-1}, s_1^{j-1}, m, t) P(s_j|a_1^j, s_1^{j-1}, m, t) \qquad (2.5)$$

m is the length of s, j is a sentence position in s, s_1^j is the string of source words covering up to position j, and a_1^j are the links to t for the source positions up to j. The equation states that given a target sentence, the probability of a source sentence and an alignment between this and the target sentence can be calculated based on three probabilities; (1) the probability of the length of the source sentence given the target sentence, (2) the probability of a link given previous links, previous source words, the length of the source sentence, and the target sentence, and (3) the probability of a source word given previous and the current link, previous source words, the length of the source sentence, and the target sentence. From these parts, the following story on how translation is done can be extracted:

> **Translation story for model 1 and 2**
> A translation is created by first determining its length, then filling the sentence positions one at a time by first determining which original word the position links to, and then determining which word form to insert.

In fact, model 1 and 2 do not condition on as many events as equation (2.5) provides for. Both models assume all sentence lengths (of reasonable lengths) to be equally probable. The first probability is therefore a constant value. The difference between the two models lies in the link modelling. Model 1 assumes that all links are equally probable, i.e. linking the first source word to the last target word is for example just as probable as linking

[2] In this thesis, a word in translation is thought to be any characters separated by white space.

it to the first. Model 2, on the other hand, assumes that the probability of linking a source position to a target position is conditioned on the source position and the length of the sentences. For both models, the probability of choosing a word form is only conditioned on the word form it stems from.

IBM model 3, 4 and 5. The second story is expressed in the simplified[3] equation (2.6):

$$P(s, a|t) = \prod_{i=1}^{l} n(\phi_i|t_i) \prod_{j=1}^{m} tr(s_j|t_{a_j}) \prod_{j=1}^{m} d(j|a_j, m, l) \tag{2.6}$$

In addition to the previously described notation, ϕ is the *fertility* of a target word, i.e. that number of words it translates into. i is a sentence position in the target sentence, which has length l. In other words, equation (2.5) has been recast with some new probabilities; (1) n, the probability of how many words a target word translates into given the word form, (2) tr, the probability that a source word form is the translation of a target word form (the same as in the previous), and (3) d, the *distortion* probability, i.e. the probability that the word form appears in a source sentence position given the target sentence position that it links to, and the lengths of the sentences. In plain words, this equation tells the following generative story:

> **Translation story for model 3, 4 and 5**
> A translation is created by first determining how many words each original word translates into, and which word forms these have, then secondly, determine at which sentence positions to place the word forms.

The difference between model 3 and 4 lies in the distortion model. Instead of modelling reordering between absolute sentence position, model 4 models relative movement. It is for example able to model that a given translation of a target word will have a tendency to move one position to the left of the translation of the previous target word. This also means that

[3]For reasons of clarity, we do not include factors that handle extraneous words that cannot be accounted for by the original words, and factors that handle the fact that the same alignment can appear via multiple paths.

model 4 conditions its linking decision on its previous linking choice, while model 3 only includes sentence lengths.

Model 5 tries to mend the problem that the previous two models are *deficient*, i.e. they waste a lot of probability mass on impossibilities. A small amount of probability is for example wasted on the alignment that links all words to the same position. That is, all words in the translation might appear on top of each other in third sentence position. In addition, the relative distortion of model 4 is unaware of sentence boundaries, therefore probability is wasted on movement out of the sentence. When it e.g. models the probability of movement for the word in first position, then left movement will be considered even though it is an impossibility.

Training the IBM models. All the information needed to calculate the parameters of the IBM models are present in a word alignment. Using maximum likelihood estimation, it is possible to calculate how many words a given word links to, how often it links to a given word form, and how often a given sentence position links to another sentence position.

Unfortunately, unlike the training material for language models, which is plain text, word alignments do not occur as products of unrelated activities. The closest source available is parallel corpora without the alignments. Brown et al. (1993) solve this by stepwise estimating the parameters and the alignment at the same time. For this, they use the *expectation maximization* (EM) algorithm (Dempster et al., 1977).

The EM algorithm works in two steps; first, the expectation step estimates the probability of all alignments between parallel sentences based on the current parameters, then the maximization step normalizes over the entire data set to update the parameters. These two steps are repeated until a predetermined threshold is reached.

Because the models are increasingly complex, the lower models are used to bootstrap the higher models. This is both due to the computational complexity, but also because the EM algorithm is only guarantied to reach a local maximum. This means that the initial parameter guess has direct influence on which optimum is reached. By getting these from more simple models, the intention it to get better initial guesses when calculating the harder models. The parameters from the previous model is used to start up training of the next. Since the algorithm needs a set of parameters to

make the estimation, initial parameters are given uniform values, i.e. all alignments are equally probable.

Problems with the IBM models. An unfortunate aspect of the IBM model alignments is that they are asymmetrical. That is, as the generative story tells, each source word originates from a target word. A source word can therefore only link to a single target word. Such a representation can not provide a satisfactory description of the word alignment example earlier in figure 2.2 on page 18, where we need many words to link to many words in order to get a suitable alignment.

In addition, some words do not link to anything. The IBM models handle this by introducing an imaginary NULL word initial in the target sentence. Source words that do not correlate with anything in the target sentence, can link to this dummy position. On the other hand, target words that do correlate with any source words, will have a high probability for zero-fertility.

In a thorough investigation of the IBM models, Och & Ney (2003) verify that the more complex models significantly improve the alignment quality over the simpler models. This is especially the case when the alignment model is first-order as in model 4 and 5, where the choice of alignment conditions on the previous choice. We will look further into improving the quality of word alignments supplied by the IBM models in chapter 4, and on this note, we leave the quirky noisy channel use of target and source, where target translates into source.

2.2.2 Phrase-based SMT

The most groundbreaking advancement from the IBM models has been a change in the basic units of translation from single words to sequences of words. Interestingly enough, this idea was already mentioned by Warren Weaver in his famous memorandum in connection with the problem of word sense disambiguation[4]:

> The difficulty you mention concerning Basic seems to me to have a rather easy answer. It is, of course, true that Basic puts

[4] The *Basic* mentioned in the quote is a simplified version of English supposed to contain the core of the language.

> multiple use on an action verb such as *get*. But, even so, the two-word combinations such as *get up*, *get over*, *get back*, etc., are, in Basic, not really very numerous. Suppose we take a vocabulary of 2,000 words, and admit for good measure all the two-word combinations as if they were single words.
>
> (Weaver, 2003 [1949], p.14 (authors' emphasis))

This sequence-based approach to SMT is best known as *phrase-based* SMT (PSMT) (Zens et al., 2002; Koehn et al., 2003). This is a potentially misleading name, since the term 'phrase' has a long tradition in linguistics as a sequence of words that functions as a single unit in the syntax of a language. Here, we will nevertheless follow the SMT traditional and use *phrase* to mean any consecutive sequence of words. When referring to the linguistic meaning, we will use the term *syntactic phrase*.

The strength of this approach may lie in its ability to handle collocational relations within the sentence. This makes it very compatible with the frequency-based nature of human language. To a large extent, words prefer occurring together with given other words. In its extreme form, this is known as idioms and selectional restrictions, that is, when the probability of certain words occurring together is very high. But in general, this property is found in most of language, though with less radical probabilities. This property is also seen in the success of the n-gram language model. The PSMT approach therefore owes a lot of its success to its surface-near nature.

The phrase-based approach overcomes many of the weaknesses of the word-based IBM models. The ability to directly describe correspondences between strings of different lengths improves modelling considerably. This provides a direct relation between the source context and the target word selection. In addition, the phrases provide a much better model for local reordering than the distortion probability. Finally, in translation, some content-weak words do not carry over to the other language. This means that some source words seem to be deleted in translation, and some target words seem to appear out of nowhere. The IBM models treat this in a very chaotic way, by letting some words link to a non-existing NULL word, and by trying to throw in unmotivated words in target sentence

generation. This no longer constitutes a problem, since the phrase correspondences hold between different length word sequences. That is, these deletions and insertions are motivated lexically by their context, instead of appearing sporadically.

At this point, we will introduce a relevant development in connection with PSMT, namely the reformulation of the fundamental equation of SMT (equation (2.2) on page 16), within the framework of maximum entropy. Och & Ney (2002) propose a direct modelling of the translation process based on the maximum entropy approach as described in (Berger et al., 1996). Instead of employing the Bayesian Theorem, the posterior probability $P(t_1^I|s_1^J)$ is expressed as a set of M submodels $h_m(t_1^I, s_1^J)$, which is each weighted by a model parameter λ_m:

$$P(t_1^I|s_1^J) = \frac{exp\Big(\sum_{m=1}^{M} \lambda_m h_m(t_1^I, s_1^J)\Big)}{\sum_{t'^I_1} exp\Big(\sum_{m=1}^{M} \lambda_m h_m(t'^I_1, s_1^J)\Big)} \tag{2.7}$$

Since the normalizing denominator only depends on the source sentence, it has no influence on the maximization of the entire probability. This leads to the following equation:

$$\hat{s}_1^I = \underset{e_1^I}{\operatorname{argmax}}\Big\{P(t_1^I|s_1^J)\Big\} \tag{2.8}$$

$$= \underset{e_1^I}{\operatorname{argmax}}\Big\{\sum_{m=1}^{M} \lambda_m h_m(t_1^I, s_1^J)\Big\} \tag{2.9}$$

This model is also known as the *log-linear model*, and an interesting aspect is that the noisy channel approach can be viewed as a special case that contains two submodels, the language and the translation model, that are weighted equally.

A major advantage of this generalization is that it provides a basis for easily enriching the translation process with additional information. When a new phenomenon related to the translation process is modelled, it is merely added to M with an appropriate weight conveying its importance.

In addition to this, there is no theoretical basis for weighting the models unevenly in the noisy channel model. Tillmann (2001), however, improve translation quality by applying model scaling factors to the language model and the distortion model. This behavior is an essential part of the log-linear model, and therefore easily accounted for within this framework.

Even though the original formulations of PSMT were noisy channel models (Zens et al., 2002; Koehn et al., 2003), they employed model weighting, which in essence makes them log-linear models. We therefore choose to view them as such. The description will be based on (Koehn et al., 2003), which represents the most widespread notion of PSMT partly due to the Pharaoh decoder (Koehn, 2004a) and other freely available tools released by Philipp Koehn. The description extends the original (Koehn et al., 2003) slightly, so that it covers the system that was used as baseline for the NAACL 2006 workshop on SMT (Koehn & Monz, 2006), since this toolkit will act as baseline for our experiments. Throughout the thesis, we will call this approach *traditional* PSMT. First, we will look at the submodels of PSMT, and then, we will look at how phrase pairs are extracted.

The models of phrase-based SMT

Many of the aspects of translation that were modelled explicitly in the IBM models, are now modelled implicitly in the phrase pairs. Instead of a fertility model and a word translation model, a *phrase translation model* is introduced. A *distortion model* is still needed, but its task is somewhat different, and it is therefore redefined. Koehn et al. (2003) also use a *lexical weighting model* that evaluates the phrases internally, a *word penalty* that biases against or towards making longer sentences, and a *phrase penalty* that biases against or towards using longer phrases. We will describe these models in the following. The language model remains the same, and it will not be described here.

- *Phrase translation model:* Given the availability of a phrase correspondence $(\bar{t}, \bar{s})$, its probability is calculated by maximum likelihood estimation:

$$P(\bar{t}|\bar{s}) = \frac{count(\bar{t}, \bar{s})}{\sum_{\bar{s}} count(\bar{t}, \bar{s})} \tag{2.10}$$

This is calculated for both translation directions, meaning that the system will contain the two phrase translation models, $P(\bar{t}|\bar{s})$ and $P(\bar{s}|\bar{t})$.

- *Distortion model:* In PSMT, the importance of the distortion model has decreased, since a lot of local reordering is treated by the phrase pairs. Therefore, a simple model was employed that merely punishes the reordering of phrases.

$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|} \tag{2.11}$$

  a_i is the start word position of the source phrase that is translated into the ith phrase position of the target sentence, and b_{i-1} is the end word position of the source phrase that is translated into the $i-1$th position. This merely states that if words that were adjoining in the source sentence, are no longer adjoining in the target sentence, then the translation is punished according to the length of the move. The distortion can be restricted by a distortion limit, so reorderings over a given length are rejected.

- *Lexical weighting model:* Another way of evaluating a phrase pair is by looking at the probability of its internal alignment. First, a word translation probability distribution is calculated from a word alignment of the parallel corpus using maximum likelihood estimation parallel to equation (2.10). Then, each phrase pair is evaluated by the following equation:

$$P(\bar{t}|\bar{s}, a) = \prod_{i=1}^{n} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} p(t_i|s_j) \tag{2.12}$$

  where a is the word alignment of the phrase pair $(\bar{t}, \bar{s})$, $\bar{t}$ covers position $i = 1,\ldots,n$, and $\bar{s}$ covers position $j = 0,\ldots,m$ (position 0 is the non-existing NULL word). In plain words, the equation gets a combined value for the phrase based on the strength of its links as evaluated by the word translation probabilities.

- *Word penalty:* This model either rewards or punishes each time a target word is added. This helps if one language often produces

shorter/longer sentences than the other.

- *Phrase penalty:* This model either rewards or punishes each time a phrase pair is used. This leads to the use of either more long more short phrases. This model is, for example, able to promote the use of longer phrases, and thereby strengthen the advantages of PSMT.

Phrase extraction

The entire setup mentioned above depends on the ability to extract phrase pairs from a training corpus. This was first introduced in a somewhat different framework called the alignment template model (Och et al., 1999; Och & Ney, 2004), where information on the alignment between phrases was included, but the basic units of translation were still words.

As mentioned, one problem with the IBM model alignments are that they are asymmetrical, i.e. they only allow many-to-one[5] alignments. Och et al. (1999) remedy this by introducing *symmetrization*. Two alignments are produced independently of each other with each language as source. This way both a many-to-one and a one-to-many alignment are available. These two alignments are merged heuristically, resulting in a symmetrical many-to-many alignment.

Koehn et al. (2005) describes the *grow-diag-final* (*GDF*) algorithm based on (Och & Ney, 2003), which is used to symmetrize the alignment used for the baseline system in our experiments. This process is exemplified by the alignment matrices in figure 2.3. On the left, we have the two initial, asymmetrical alignments. Then the backbone of the alignment, the intersection of the two original alignments, is extracted. This is exemplified by the black links in the center matrix. The grey links are links not shared by the alignments. Then, the intersection is extended in two steps. First, the *grow-diag* step extends the alignment to links neighboring an existing link. Each link in the matrix is surrounded by 8 neighbor links (unless it is a border link), i.e. the neighboring links of a given link in the matrix are all the links touching up to it. The algorithm only adds links that were in

[5] Since we have left the noisy channel use of source and target, many-to-one is here intended to mean that source words can each have up to many links, while target words can only have up to one link. That is, the the alignments are not necessarily many-to-one. It is a maximum.

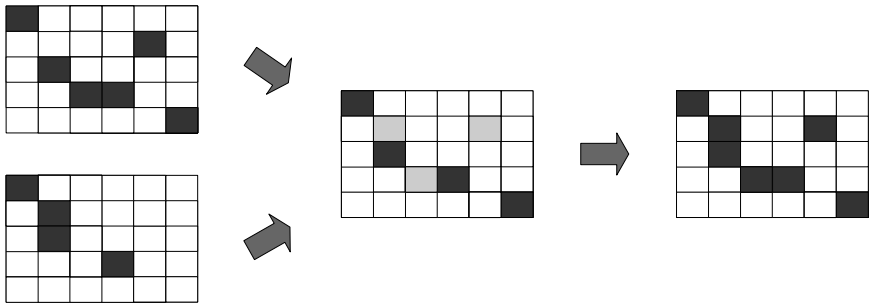


Figure 2.3: Example of GDF symmetrization. The x-axis represents the words of the source sentence, and the y-axis represents the words of the target sentence. Each square represents the link between a source and a target word.

one of the original alignments (the grey links), and it keeps adding until no grey link neighbors a black link, i.e. either an original black link, or a grey link that has been turned black earlier in the process. This step adds the two leftmost grey links in the center matrix. The second step is the *final* step. This step looks at all remaining grey links in the matrix, and adds them if either the source or the target word is not linked to anything. This step adds the final grey link, resulting in the symmetrized alignment on the right.

The phrase correspondences are extracted from the symmetrical alignment by a simple algorithm stating that any consecutive string of source words corresponds to the string of target words it links to, as long as this string is consecutive and does not link outside the source string.

This is exemplified by figure 2.4. Given the alignment that has been assigned to the parallel sentences, the light grey area of the left matrix shows the good translation phrase '*did not* → *ikke*'. Here, the included words only link to each other. The light grey area of the right matrix is on the other hand not a legal phrase, since the included word '*relax*' links to the excluded word '*af*'.

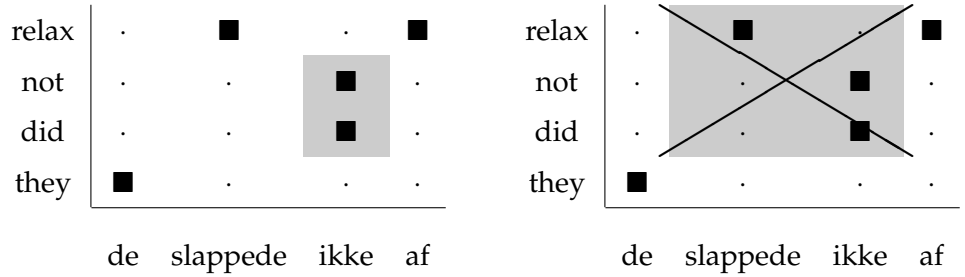


Figure 2.4: Example of a good and a bad translation phrase.

2.3 Linguistics in phrase-based SMT

One issue that has been discussed since the introduction of SMT, is its lack of linguistic knowledge. First of all, many aspects of language seem to require this. It is unclear that a task such as reordering can be handled satisfactory without some sort of higher linguistic knowledge. Secondly, one would expect a more generalized system that would make sparsity less of an issue.

Integrating abstract linguistic knowledge into SMT is not without its problems. Several, highly ambitious experiments have been made, integrating different levels of linguistic knowledge, but without the expected substantial improvements to the simple word-sequence-based approach. Especially, the incorporation of syntactic structure has been the subject of many experiments.

In this section, we will describe two approaches that have integrated linguistic knowledge in PSMT. These are meant to provide an impression of the possibilities in doing this by attacking the problem from different linguistic levels. First, we describe the idea of *factored translation models* that opens for integration of additional linguistic information pertaining to and below word level. From the sub-word level, we move to the sentence level with the concept of *hierarchical phrases* that integrates hierarchical sentence structure in the decoding process.

2.3.1 Factored translation models

Koehn & Hoang (2007) present the idea of factored translation models. They supply additional word-related information to a PSMT system. Instead of translating surface word phrases, the basic unit is a vector annotated with several levels of information for the words of the phrase. They include information such as lemma, morphology, part-of-speech, and statistical word classes.

The phrase-based approach is also extended with a generation step. In its most extreme form, this allows the system to translate without surface form information. Instead, the abstract levels are translated, and then a surface form is generated from these using purely target-side operations. The generation models can therefore be trained on monolingual, target data, which is much easier to come by.

Koehn & Hoang do not provide a systematic test of the approach. The data sets tested on are not from the same language, and the additional information used differs between experiments. Nevertheless, their experiments indicate that translation is improved when data is sparse (20,000 – 52,000 sentences of parallel training data), an effect that wears of when moving to larger amounts of training data (750,000 sentences).

Manual analysis of the large English-German task, however, shows a substantial decrease in noun phrase agreement errors. That is, the approach seems to help when translating the case-unspecified English noun phrases into the case-specific German. This, not surprisingly, indicates that an effect of the approach relies heavily on a difference in morphological complexity between the languages involved.

This approach is implemented in the open source decoder, *Moses*, that was developed at the 2006 Johns Hopkins University summer workshop (Koehn et al., 2007).

2.3.2 Hierarchical phrases

Chiang (2005, 2007) extends PSMT to contain *hierarchical phrases*, which are template phrases with variables that can be instantiated by other phrases. An example of these template phrases between Chinese and English is:

$$\text{yu } X_1 \text{ you } X_2 \rightarrow \text{have } X_2 \text{ with } X_1$$

These composite phrases bring a higher level of generalization to the original PSMT approach. This reduces the sparsity problem and creates a basis for much longer phrases. According to Koehn et al. (2003), when training a traditional PSMT system on parallel corpora of up to 20 million words, the gain from learning phrases longer than 3 words is small. In effect, the hierarchical phrases extend the local reordering advantages of phrase-based SMT to a much wider area. In his experiments, Chiang handles phrases covering up to 10 source words. He does not examine where the limit goes, but preliminary experiments show that phrase lengths as long as 15 still brings improvement.

In essence, the approach does not contain linguistic information, but the hierarchical nature of the phrases is a step in that direction. Chiang calls it a formally, but not linguistically, syntax-based approach. He does experiment with biasing towards linguistic phrases, but this does not bring improvement. In a similar approach, (Marcu et al., 2006) show improvements based on a linguistically syntactic structure.

2.4 Evaluating machine translation

Being able to evaluate a machine translation system is crucial. It provides a goal to strive for, without which the exercise would be pointless. However, evaluation of machine translation has proven to be quite difficult.

For the most part, bilingual language users are able to perform an intuitive evaluation on whether a translation is good or bad, or whether one translation is better than another. However, these judgements are subject to a high level of variance due to many aspects such as expected purpose of the translation (e.g. overview, comprehension, publishing), the evaluators' level of conscious linguistic knowledge, and whether the evaluator weighs content or grammar highest.

Attempts have therefore been made at increasing inter-subjectivity by casting human evaluation within the boundaries of well-defined judgement tasks. These human evaluation tasks provide the best insight into the performance of an MT system, but they come with some major drawbacks. It is an expensive and time consuming evaluation method, and therefore it is less suited for tasks like everyday assessment of system progress or

testing out new ideas.

To overcome some of these drawbacks, automatic evaluation metrics have been introduced. These are much faster and cheaper than human evaluation, and they are consistent in their evaluation, since they will always provide the same evaluation given the same data. This is opposed to human evaluation where the intra-annotator agreement is far from perfect (Callison-Burch et al., 2007).

The disadvantage of automatic evaluation metrics is that their judgments are often not as correct as those provided by a human. The optimal metric should mirror the judgements of humans.

In short, MT evaluation is the task of scoring a translation given its source. This is, in fact, precisely the task statistical machine translation is seeking to solve. Here the MT system evaluates a variety of hypothesis translations created from the input. It then chooses the best translation based on this evaluation. In this light, you need to solve the problem of MT to solve the problem of automatic MT evaluation.

The evaluation process, however, has the advantage that it is not tied by the realistic scenery of translation. Most often, evaluation is performed on sentences where one or more gold standard reference translations already exist. So where the SMT system needs to piece its evaluation together from relevant references in its vast experience, the automatic evaluation metric possesses a fixed set of gold standard translations for reference that are targeted at exactly this translation task.

Nevertheless, even a large amount of gold standards will in reality not be enough to fully cover the potential variation leading to acceptable translations. This means that even though automatic evaluation has better premises, perfect evaluation still faces the same barriers as SMT in that it is necessary to evaluate based on an inadequate data set.

Automatic MT evaluation therefore settles for high statistical correlations with human judgements over large amounts of data. This evens out the noise brought on by the imperfections of automatic evaluation, but it also rules out certainty in evaluation of single sentences.

In the following, we will describe a variety of the methods currently used in human and automatic evaluation of MT. We start out with human evaluation.

2.4.1 Human evaluation

Jones et al. (2005) describe a comprehension test for evaluating machine translation. Callison-Burch et al. (2007) examine three approaches to human evaluation: 1) scoring the translations based on their adequacy and on their fluency, 2) ranking the translations relative to each other, and 3) ranking the translation of syntactic constituents relative to each other. In this section, we will describe these approaches as representative for human evaluation.

Comprehension testing

Jones et al. (2005) describe a reading comprehension test based on the Arabic Defense Language Proficiency Test. This is a test used by the U.S. Defense Department to examine a linguist's proficiency in handling real-world tasks. The subjects are asked to answer a set of questions based on information from Arabic texts.

In the MT modified comprehension tests, rather than supplying the subjects with Arabic texts, they are given the machine translated output from the Arabic texts. The system is then scored based on how well the subjects are able to answer the questions, and the amount of time they use.

Scoring adequacy and fluency

The judges are presented with a gold-standard sentence and some translations. Table 2.1 shows the scales used for evaluation when the language being translated into is English. Using this scale, the judges are asked to assign a score to each of the presented translations.

Accuracy and fluency is a widespread means of doing manual evaluation. This in part owes to the fact that it is used in the influential NIST MT workshop that is held annually to evaluate how state-of-the-art MT is doing (LDC, 2005).

Callison-Burch et al. (2007) show that there are some problems with this approach. First of all, judges do not show an impressive amount of agreement with each other or with themselves when measured using Cohen's Kappa (e.g. Bakeman & Gottman, 1997).

Adequacy	**Fluency**
How much of the meaning expressed in the gold-standard translation is also expressed in the target translation?	How do you judge the fluency of this translation?
5 = All 4 = Most 3 = Much 2 = Little 1 = None	5 = Flawless English 4 = Good English 3 = Non-native English 2 = Disfluent English 1 = Incomprehensible

Table 2.1: Scales used for human evaluation of adequacy and fluency (LDC, 2005).

Cohen's Kappa is a non-parametric agreement measure that compares the actual agreement of two judges with what might be expected by chance. It is defined as follows:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.13}$$

$P(A)$ is the amount of agreement between judges, and $P(E)$ is the agreement expected to occur by chance. A κ values of 1 is full agreement, and 0 is no agreement. $P(A)$ is given by the total number of agreeing judgments, divided by the total amount of judgements. $P(E)$ is given by the amount of agreement that might be expected, divided by the total amount of judgements. The expected amount of agreement is calculated with respect to the distribution of scores. If certain scores are used more frequently by the judges than other scores, then they are more likely to agree by chance, than

Kappa statistic	**Strength of agreement**
< 0.00	Poor
$0.00 - 0.20$	Slight
$0.21 - 0.40$	Fair
$0.41 - 0.60$	Moderate
$0.61 - 0.80$	Substantial
$0.81 - 1.00$	Almost perfect

Table 2.2: Interpretation of κ scores as suggested by (Landis & Koch, 1977).

if the scores are evenly distributed. In other words, agreement is least likely by chance when scores are evenly distributed.

The interpretation of the resulting κ scores used by (Callison-Burch et al., 2007) is given in table 2.2. Other interpretations set higher demands. For example, (Krippendorff, 1980) claims that κ score should be over 0.67 to warrant tentative conclusions to be drawn. We follow (Callison-Burch et al., 2007) in using table 2.2.

In (Callison-Burch et al., 2007), adequacy and fluency judgements respectively get κ scores of 0.226 and 0.250 for inter-annotator agreement and 0.468 and 0.537 for intra-annotator agreement. These are by no means impressive agreement levels. It is especially surprising that people agree so little with themselves.

These values should also be seen in the light that the expected level of agreement is uniformly set. The probability of assigning each of the five scores is set to $\frac{1}{5}$. This assumes that each of the five scores are used equally, which seems highly unlikely, since judges may have a tendency to avoid extreme scores (1 and 5). As mentioned above, choosing an even distribution has the effect that $P(E)$ is at a minimum, which in turn yields the maximal

κ score. In other words, the agreement reported in (Callison-Burch et al., 2007) may in fact be too optimistic.

An additional problem with adequacy and fluency is that there is an almost perfect correlation between the two judgements. This indicates that judges are not able to distinguish the two (Callison-Burch et al., 2007).

Ranking sentences

This is a fairly simple evaluation method. Given the source sentence, the judges are asked to evaluate a number of translations by ranking them relative to each other. They are allowed to evaluate more translations as equally good.

According to (Callison-Burch et al., 2007) this approach yields both greater inter-annotator agreement (κ = 0.373) and intra-annotator agreement (κ = 0.623).

Ranking constituents

Callison-Burch et al. (2007) also introduce a novel approach to human evaluation. The approach is basically the same as for sentence ranking with the exception that only part of the sentence is evaluated. A syntactic constituent is highlighted in the source sentence, and the equivalent words are highlighted in the translations. The judges are then asked to rank the translations relative to each other solely based on the highlighted part.

This approach achieved the highest agreement in their experiments with an inter-annotator agreement of κ = 0.540 and intra-annotator agreement of κ = 0.738.

Human evaluation in the thesis

In the manual evaluations in chapter 6, we will follow the sentence ranking approach. The main reason for this choice is the lower inter-annotator agreement of adequacy and fluency, and the indications that judges have problems distinguishing the two. In addition, even though the constituent ranking approach shows higher inter-annotator agreement than sentence ranking, we do not employ this approach, since it is not 100% clear to us

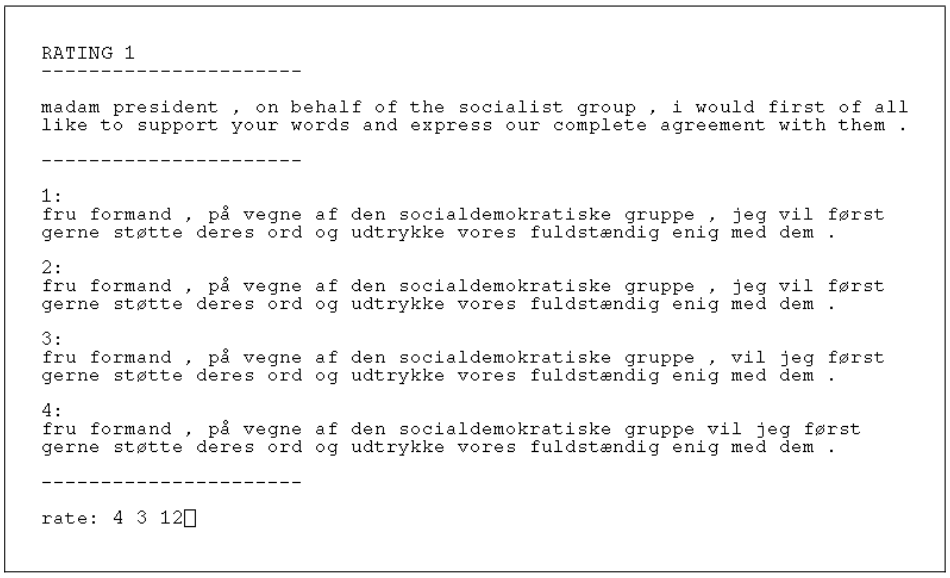


Figure 2.5: Screenshot of the human evaluation scenario.

what conclusions can actually be drawn from it. There are many restrictions on which constituents can be evaluated, and in the end, we are not sure this leads to an evaluation of the sentence. Finally, constructing a comprehension test for Danish and Arabic is a major operation beyond the scope of this thesis, we therefore did not employ this test.

Figure 2.5 is a screenshot of the interface the human evaluators were presented with. First, the evaluators can see where they are in the process (here, at the first rating). Then the source sentence is presented, followed by translations from four different systems, and a command prompt telling the evaluator to rate. The order the translations appear in is randomized for every rating.

To keep the number of ratings down, the reference sentence was not presented for evaluation. This might have been an interesting comparison, but on the other hand, the reference would probably stand out, since the machine translated sentences where often very similar as in the example. This might draw a lot of attention away from the actual purpose of the evaluation. We also restrict the sentence length to a maximum of 30, since Europarl has very long sentences that are extremely hard to process.

The evaluators are asked to rank the four translations given the source

sentence. Ties are allowed. This is exemplified by figure 2.5, where the human evaluator has rated translation 4 as best, followed by translation 3, and finally, translation 1 and 2 are rated as tied worst. This is indicated by grouping 1 and 2 in the rating.

The rankings provided by the evaluators is converted to a *fractional rank*. This means that the above rating would have produced the following scores for the four translations: 3.5 3.5 2 1. Here, translation 1 and 2 are both given score 3.5, since they share rank 3 and 4. If all translations were judged equal, they would each get the score 2.5, since they are sharing rank 1, 2, 3, and 4. The advantage of this approach is that even though translation 4 gets rank 1 in both scenarios, it gets a better score when it outperforms the others, than when it ties with them. We then determine the average score by taking the median of all scores assigned to a system. As a reference, we also provide the mean.

When using more than one evaluator, we employ *Cohen's kappa* as described earlier in this section to test their level of agreement, and significance of the results is tested using the *Wilcoxon signed-rank* test as described by Siegel (1956). This is a non-parametric test for two related samples. It compares differences in the scores assigned to the systems. The absolute values of differences are ranked after size with lowest rank to smallest difference. Zero differences are excluded. If more absolute differences are tied, they are all assigned a fractional rank. Based on these ranks, two sums are produced. One for all the observations where system 1 is best ($T_{(+)}$), and the same for system 2 ($T_{(-)}$). The zero hypothesis assumes that these T values are equal, i.e. the two systems do not differ. If the number of observations n exceeds 15 (after zero differences have been removed), the distribution tends towards the normal distribution. A z value can be calculated from the lowest of the T values using the following equation:

$$z = \frac{T - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}} \tag{2.14}$$

Because of the low number of rating options, our data will exhibit a lot of ties. According to Siegel (1956), ties affect the T value, but the effect is small and unimportant in practice.

2.4.2 Automatic evaluation

The importance of human evaluation cannot be exaggerated. It provides the most correct and detailed picture of the performance of an MT system. Nevertheless, for some tasks, automatic evaluation is indispensable. In situations like everyday system evaluation, human evaluation can be too expensive, slow, and inconsistent. Therefore, an automatic evaluation metric that is reliable, is very important to the progress of the field.

In this section, we will describe the most widely used automatic evaluation metrics at the moment, BLEU, NIST and METEOR. A lot of other interesting metrics have been proposed, and according to (Callison-Burch et al., 2007) some of these such as *Semantic role overlap* (Giménez & Màrquez, 2007) and *ParaEval-Recall* (Zhou et al., 2006) show greater correlation with human evaluation. We will, however, not go any further into these in this thesis. Instead we have chosen to focus on the more commonly used metrics.

BLEU

The BLEU metric (Papineni et al., 2002) compares n-gram overlap between a translation and possibly multiple references. This is done based on the *modified n-gram precision*, which is calculated by dividing the number of n-grams in the translation that match an n-gram in a reference, by the total number of n-grams in the translation. This is called modified, since each reference n-gram is only allowed to match once.

The BLEU score is measured on document level, not sentence level. This means that for a given n, modified precision is the total number of matching n-grams in the document divided by the total number of n-grams in all translations. This is formalized by equation (2.15).

$$precision_n = \frac{\sum_{T \in Translations} \sum_{n\text{-}gram \in T} match(n\text{-}gram)}{\sum_{T' \in Translations} \sum_{n\text{-}gram' \in T'} count(n\text{-}gram')} \tag{2.15}$$

While it makes perfect sense to compare how much of a translation is found in any reference (precision), it makes less sense to examine how much of all the reference translations is present in the translation (recall). If

for example the same meaning is expressed in four different ways in four references, then the translation will get full credit for hitting one of these in precision, but since only one of four is present in the translation, recall will be bad. Recall therefore seems inappropriate with multiple references.

For these reasons, the BLEU metric avoids recall and instead introduces a heuristic *brevity penalty* (BP). This penalty punishes too short translations, which will otherwise have a tendency to get higher precision due to fewer total n-grams. That is, the BP acts as a counterweight to the modified precision.

The BP is based on the reference that is closest in length for each translation. Then summing over the entire corpus, a total length of all references (r) and all translations (t) is found. BP is then calculated by equation (2.16).

$$BP = \begin{cases} 1 & \text{if } t > r \\ e^{(1-r/t)} & \text{if } t \leq r \end{cases} \tag{2.16}$$

The BLEU score is calculated by taking the geometric mean of the modified precision for N's up to a maximum n-gram length and multiplying it by the BP as in equation (2.17). The maximum n-gram length is usually set to $N = 4$, and n-grams are usually weighted equally $w_n = 1/N$.

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log precision_n\right) \tag{2.17}$$

The BLEU metric has been one of the most influential additions to the field of MT in the new millennium. This both for good and for bad.

BLEU was the first automatic evaluation metric proven to display a high level of correlation with human evaluations (Papineni et al., 2002; Doddington, 2002). This was exactly what the community was looking for, and BLEU quickly became the standard evaluation metric in machine translation. This status is for example reflected in the fact that it is the official evaluation metric for the influential NIST Open Machine Translation Evaluation.

Finding this standard evaluation metric helped push the field. Now people had a subjective, fast and cheap evaluation option that it was feasible to optimize system parameters against. This made a lot more research pos-

sible, and less time was used on evaluation.

It is, however, uncertain to what extent this metric has controlled the direction of machine translation. If everybody is using the same metric, and this metric biases in favor of certain directions, then these directions will get more attention. To a certain extent, BLEU seems to contain such biases. As an example, it has been shown that BLEU evaluates a rule-based system like SYSTRAN unfairly low compared to statistical systems (Callison-Burch et al., 2006; Callison-Burch et al., 2007).

The main problem with BLEU may not as much be the metric itself, as it is the way people utilize it. BLEU stands for *Bilingual evaluation understudy*, and as the word 'understudy' signal, the metric was meant as a supplement to human judges. In a critical evaluation of BLEU, Callison-Burch et al. (2006) report that BLEU not necessarily correlates well with human evaluation, and they suggest that BLEU should only be used to compare systems with 'similar translation strategies' and as an 'objective function' for system optimization.

NIST

The NIST metric (Doddington, 2002) is an extension of the BLEU metric. The introduction of this metric tried to meet two characteristics of BLEU. First, the geometric average of BLEU makes the overall score more sensitive to the modified precision of the individual n's, than if the arithmetic average is used. This may be a problem if not many high n-gram matches exist. Second, all word forms are weighted equally in BLEU. Less frequent word forms may be of higher importance for the translation than for example high frequent function words, which NIST tries to compensate for by introducing an *information weight*. Additionally, the BP is also changed to have less impact for small variations in length.

The information weight of an n-gram *a b c* is calculated by the following equation:

$$info(abc) = \log\left(\frac{count(ab)}{count(abc)}\right) \tag{2.18}$$

This information weight is used in equation (2.20) instead of the actual count of matching n-grams. In addition, the arithmetic average is used

instead of the geometric, and the BP is calculated based on the average reference length instead of the closest reference length. The lengths of these are summed for the entire corpus (r) and the same for the translations (t).

$$BP = \exp\left(\beta \cdot \log\left(\min(\frac{t}{r}, 1)\right)\right) \tag{2.19}$$

$$NIST = BP \cdot \sum_{n=1}^{N}\left(\frac{\sum_{n\text{-}gram \in matching} info(n\text{-}gram)}{\sum_{n\text{-}gram' \in translation} count(n\text{-}gram')}\right) \tag{2.20}$$

N is usually set to 5, β is set so BP is 0.5 if the translations length is 2/3 of the references.

The NIST metric is very similar to the BLEU metric, and their correlations with human evaluations are also close. Perhaps NIST correlates a bit better with adequacy, while BLEU correlates a bit better with fluency (Doddington, 2002).

METEOR

The METEOR metric (Banerjee & Lavie, 2005; Lavie & Agarwal, 2007) was created to overcome some of the weaknesses of BLEU such as a lack of recall and the missing ability to judge on a sentence level. Experimental results indicate that METEOR is a superior metric to BLEU (Banerjee & Lavie, 2005; Callison-Burch et al., 2007).

In METEOR, a word alignment is produced between the translation and the references. Only one-to-one alignments are allowed, and the final score is based on the best matching reference. The alignment is created based on exact matching word forms, as was the case for BLEU and NIST, and in addition, morphological stemming and synonymy relations.

Based on the alignment, a weighted F-score is calculated. Precision is matching unigrams divided by total unigrams in translation, and recall is matching unigrams divided by total unigrams in reference. The weighted F-score means that either precision or recall can be optimized to greater importance.

$$F = \frac{precision \cdot recall}{\alpha \cdot precision + (1 - \alpha) \cdot recall} \tag{2.21}$$

Finally, the F-score is assigned a penalty based on how fragmented the word alignment is. In short, this is a weighted measure of how many consecutive matching sequences the translation contains. One long sequence is better than many short ones. This means less reordering between the translation and the reference. The penalty is weighted by two parameters; γ for maximum penalty, and β for how much impact fragmentation has on the maximum penalty.

$$METEOR = (1 - \gamma \cdot frag^{\beta}) \cdot F \tag{2.22}$$

The three weights α, β, and γ need to be optimized, and they are highly language dependent (Lavie & Agarwal, 2007). This language dependency is extended by the need for linguistic tools for stemming and determining synonymy relation. At present, the METEOR evaluation tool is capable of evaluating English, French, German, Spanish, and Czech.

A problem with METEOR is that it is slower than BLEU, and therefore less suited as a metric to optimize towards. In addition, the set of supported languages is small, and only English is actually fully developed at the moment. And finally, the metric needs to be optimized, which means that if results are to be comparable, then a standard set of parameters is needed for at least every language or perhaps every language pair. One might anticipate that a different set of parameters is needed when evaluating translations from Chinese into English, than what is needed when the source language is Danish.

Automatic evaluation in the thesis

In out experiment, we will follow general practice in SMT and employ the BLEU metric as our main automatic evaluation metric. Our SMT systems will therefore also be optimized towards BLEU. BLEU score will be reported as %BLEU, since these are often easier to comprehend. In our experiments, we do not take casing into consideration, which in effect means that we use case-insensitive BLEU.

We also report NIST scores. These are merely meant as a point of comparison. The main weight in discussions will, however, be on the BLEU scores.

We do not report on METEOR scores. Since Danish and Arabic are not on the list of supported languages for METEOR, we are not able to evaluate our experiments using this metric.

We use bootstrapping to test the significance of the BLEU and NIST results (Zhang et al., 2004; Koehn, 2004b). Since we use the bootstrapping toolkit[6] supplied by (Zhang et al., 2004), we follow their description here.

Bootstrapping resamples the translation set into a large amount of new sets (e.g. 1000) of the same size. The sets are randomly drawn from the original set with replacements, since all new sets would otherwise be the same as the original. These 1000 new sets lead to 1000 BLEU scores. The same is done for the comparison translation set. Then the difference in BLEU score is calculated for each of the 1000 set pairs, and the 95% confidence interval for the BLEU score differences is obtained by cutting away the bottom and top 2.5% of these. If the resulting interval does not overlap with zero, the translation sets significantly differ from one another at a 95% level. We will now turn to the main subject of this thesis: *reordering*.

[6] Available at
http://projectile.is.cs.cmu.edu/research/public/tools/bootStrap/tutorial.htm.

Chapter 3

Reordering

From a formalized point of view, translation can be split into two tasks: 1) selecting appropriate words, and 2) determining the order they should appear in. This chapter is dedicated to the second task.

More precisely, this chapter is entitled reordering, because from a translation point of view, the order of the words in the source sentence is often different from that of the corresponding words in the target sentence. That is, the words need to be *reordered* as part of the translation. In this chapter, we will look at how these differences are handled in translation.

This does not mean that word reordering is independent of word selection. Two different, but perfectly acceptable, word orders may warrant different words. An example of this is the distinction between passive and active voice. Depending on the purpose of the translation, both options may be regarded as acceptable translations, since the difference mainly pertains to the level of information structuring[1]. As illustrated by example (4), the verb forms used in creating an active sentence (4a) are often not the same as those used in the corresponding passive (4b). In addition, the passive needs the preposition *by* to include the agent role supplied by *John*.

(4) a. John kicks Mary

b. Mary is kicked by John

[1] The main function of voice is assumed to be the identification of what the speaker regards as most important for the hearer to focus on in the sentence (e.g. Diderichsen & Elming, 2005; Tomlin, 1995).

As mentioned in section 2.2.2, PSMT actually utilizes the connection between word choice and word order to a certain extent, since the basic building blocks (phrase pairs) contain both word translation and reordering.

Phrases can, however, not be expected to handle all possible reordering, and traditional PSMT meets the requirement for additional reordering by letting the decoder in theory try all phrase reorderings. In order to keep control of these reorderings, the distortion model is added as a restrictive parameter.

This is a very simplistic way of handling reordering, but it has the advantage that reordering is modelled as a separate phenomenon without loosing its connection to word selection. This is the case, since the reordering model is simply one of many parameters in the log-linear model.

It is this possibility of creating a linguistically based model that focusses of the task of reordering, but still is fully integrated in the PSMT system, that is the main goal of this thesis, and this chapter will serve as the foundation.

In the following, we will first provide a language comparison of the language pairs used in the experiments. This will provide some insights into the kind of reorderings we will be dealing with in the experiments. Then we will look at how reordering is treated in PSMT; both problems of the traditional approach, and which solutions have been proposed to these problems. Finally, we point out weaknesses of these solutions. This sections also serves as the main motivation for the work done in the thesis.

3.1 Language comparison

In this section, we will take a closer look at how these differences in word order manifest themselves in actual languages. We provide analyses comparing English to Danish and Arabic respectively, since these are the language pairs our experiments will be conducted on. The first represents a language pair with little reordering, while the second language pair contains a large amount of reordering.

3.1.1 English-Danish

The two languages examined here, English and Danish, are very similar from a structural point of view. This owes to the fact that they are closely related Germanic languages. Both of them have the basic word order SVO, which means that in what is considered the unmarked or most common construction type, the main constituents of the sentence have the following order; subject, verb, object (Comrie, 1989, ch.4).

This becomes apparent in a word alignment between the two languages, which will most often display an almost one-to-one correlation. In the hand-aligned data used in our experiments, only 42% of the sentences contain reorderings (following the definition of reordering given in section 5.3). On average, a sentence contains 0.70 reorderings.

One of the main differences between English and Danish word order is that Danish is a verb-second language: the finite verb of a declarative main clause must always be the second constituent. Since this is not the case for English, a reordering rule should move the subject of an English sentence to the right of the finite verb, if the first position is filled by something other than the subject. This is exemplified by (5), where 'they' should move to the right of 'come' to get the Danish word order as seen in the gloss.

(5) Nu kommer **de** .
[Now come **they** .]
*'Here **they** come.'*

Another difference is the treatment of sentence adverbials. In Danish, the placement of a sentence adverbial depends on the status of the clause it appears in. In a main clause it is placed after the finite verb, while it is placed after in a subordinate clause. English, on the other hand, does not exhibit this behavior. Here, the placement of sentence adverbials relates to the status of the finite verb. If this is a full verb, then the sentence adverbial appears before, while it appears after an auxiliary verb. This possibly leads to differences in word order, which is illustrated by example (6).

(6) Hun sagde **kun** at hun **ikke** havde set ham .
[She said **only** that she **not** had seen him .]
*'She **only** said that she had **not** seen him.'*

	English-Danish	**English-Arabic**
Sentences containing reorderings	42%	91%
Average reorderings per sentence	0.70	2.96

Table 3.1: Comparison of the amount of reordering in the language pairs examined in this thesis (following the notion of reordering as defined in section 5.3).

Other differences are of a more conventionalized nature. For example, address numbers are written after the street in Danish (example (7)).

(7) Han bor Nygade 14 .
[He lives Nygade 14 .]
'He lives at 14 Nygade.'

3.1.2 English-Arabic

English and Arabic stem from different language families. English is a Germanic Indo-European language, while Arabic is a Semitic Afro-Asiatic language. One place where this is especially evident, is in the basic word orders of the languages. As mentioned earlier, English has the basic word order SVO. Modern Standard Arabic (MSA) on the other hand has the basic word order VSO (Maamouri et al., 2006). We here follow Habash (2007b), who describes some of the most notable differences in English and Arabic word order.

In the hand-aligned Arabic data used for later experimentation, 91% of the sentences contain reorderings with an average of 2.96 reorderings per sentence. These figures are compared to English-Danish in table 3.1. As expected, the table reveals that reorder differences are much larger between English and Arabic than between English and Danish. The English-Arabic combination contains more than four times as many reorderings per sentence.

As seen from the basic word orders, the placement of the subject is an issue when comparing English and Arabic word order. The English subject is placed in front of the verb in declarative sentences. In Arabic, things are more complicated. Even though it is a VSO language, Diab & Habash

(2006, p.38) report that the verb only appears in front of the subject 35% of the time. In another 35% of the cases, the subject appears in front of the verb, which is often due to topicalization. The remaining 30% of the time, the subject does not appear, since Arabic is a pro-drop language, i.e. it has the possibility of omitting unstressed pronominal subjects (Comrie, 1989, p.54).

The fact that Arabic exhibits this multitude of word orders, makes it even more interesting and challenging. In a case like this, where a reordering may or may not occur, probabilistic rules may prove helpful. In example (8), we illustrate the VSO word order, where the subject *'Alrjl' (the man)* appears after the verb *'ktb' (wrote)*[2].

(8) ktb Alrjl ktAbA jdydA En mEAnAp wTn+h AlmHtl
[wrote the man a book new about suffering country+his the occupied]
'the man wrote a new book about the suffering of his occupied country'

On a subsentential level, Arabic word order diverges from English within the noun phrase. An example of this is that genitives appear after the constituent they specify in Arabic (Comrie, 1989, p.208). This is also the case for possessive pronouns, which is different from English. These have status as clitics in Arabic. That is, elements that grammatically behave as independent words, but phonetically seem attached to other words (O'Grady et al., 1997, pp.139). This enclitic attachment of possessive pronouns in Arabic is also exemplified by example (8). Here, the possessive pronoun *'h' (his)* is enclitically attached to its noun *'wTn' (country)* giving it the opposite order of English.

A final example of different word order within the noun phrase is the placement of adjectives. In Arabic, they appear after the noun they modify, whereas in English they appear before. In examples (8), this is exemplified by both *'jdydA' (new)* moving to the left of *'ktAba' (a book)*, and *'AlmHtl' (the occupied)* moving to the left of *'wTn' (country)*. We will not go into the differences in definiteness expressed by the gloss, since we believe these reflect a difference in the use of definiteness rather than a difference in word order.

[2] The example is in the Buckwalter transliteration scheme (Buckwalter, 2004) with English gloss and translation

3.2 Reordering in PSMT

In this section, we will look at reordering within the framework of PSMT. First, we will look at problems with the original proposal for reordering in PSMT. Then, we show how previous approaches have sought to remedy these problems. These lead up to the final section of this chapter, where we motivate our work by highlighting some of the problems with these previous solutions.

3.2.1 Problems with reordering in traditional PSMT

As described in section 2.2.2, PSMT has two means for doing reordering; **phrase internal reordering** and **phrase external reordering**. The phrase internal reordering is when a phrase pair has been learned where the equivalent words appear in different orders. The phrase external reordering is the distortion-restricted phrase rearrangement conducted by the decoder.

The phrase internal reordering has provided SMT with a robustness in local reordering. To a large extent, this strength owes to the fact that decisions are based on highly lexicalized information from the immediate context. This, however, means that the information is not very general, and it results in the strength disappearing when encountering unknown sequence, or sequence gets too long (long distance reordering).

As an example, a general rule in French is that the adjective appears after its noun. There are, however, a lot of exceptions to this rule, where certain adjectives prefer to appear in front of the noun. For example, *'the red house'* in French is *'la maison rouge'*, but *'the little house'* is *'la petite maison'*. In the first example, the adjective *'rouge'* appears after its noun, but in the second, the adjective *'petite'* appears in front of its noun.

The lexicalized nature of phrase-based SMT makes it very good at dealing with such a phenomenon, **given it has seen the sequence previously**. If the system for example is to translate example (9), and it has not previously seen the sequence *blue house*, then these words would have to be translated separately, and the ordering of the words could not rely on the strong information of phrase internal reordering. Instead, it would have to count on the phrase external reordering.

(9) la maison bleue avec les arbres est grande .
[the house blue with the trees is big .]
'the blue house with the trees is big .'

As described in section 2.2.2, the distortion model utilized in the original PSMT is fairly simple. It assigns a penalty to a hypothesis based on the target side distance between phrases that were adjoining on the source side. If the phrases appear in the same order on both sides, then the penalty is 0. Otherwise, the penalty increases as the target side distance between the two phrases gets longer.

In short, the basic approach is to penalize reordering and rely on the target language model to force a reordering through, if this leads to much better word sequences. Returning to example (9), the system might be able to handle the reordering by relying on second hand information such as: is it most likely that *bleu* appears between *la* and *maison* or between *maison* and *avec*.

Whether the language model is able to be helpful on this question depends entirely on whether it has experienced any of the words together in its training data. If it has not, then the hypothesis with lowest distortion penalty will be chosen. If it has, it still may not be enough, since it is only a model of the target language and not the relationship between source and target. In other words, it is indirect information. It does not have a direct relation to the source sentence. This can lead to errors if a language has multiple word order possibilities under different circumstances. Especially, if the reordering relies on non-local information.

An example of this is the difference in placement of sentence adverbials in English and Danish. As mentioned in section 3.1, this is determined by the status of the verb in English and the status of the clause in Danish. For reasons of convenience, we repeat example (6) as example (10). This example shows the difficulty for a PSMT system. Since the trigram *'hun havde ikke'* is frequent in Danish main clauses, and *'hun ikke havde'* is frequent in subordinate clauses, we need information on subordination to get the correct word order. This information might be obtained from the conjunction 'that', but a trigram PSMT system would not be able to utilize this information to do the reordering in (10), since *that* is beyond the scope of *not*.

(10) Hun sagde kun at hun ikke havde set ham .
[She said **only** that she **not** had seen him .]
*'She only said that she had **not** seen him.'*

The problems with reordering in traditional PSMT become more obvious as the reordering distance grows. One reason for this is that sparsity becomes more problematic as the phrase length gets longer. This means that there is no gain from learning phrases longer than a certain maximum. According to Koehn et al. (2003) this length is three when training on up to 20M words. Having learned the translation of a string longer than this maximum is therefore too unlikely to rely on.

To achieve these long distance reorderings, it is therefore necessary to rely on random reordering controlled by the distortion model, and as mentioned, this model is merely a bias against doing reordering. It lacks a solid basis for decision in order to do global reordering. One problem is that whereas the word level approach complies well with local translation phenomena, long distance reordering most often depends on higher level linguistic information, which is absent from the distance-penalizing distortion model.

In other words, the distortion model is not a model of the reordering process. Instead, it provides damage control for the enormous amount of reorderings attempted by the decoder. As a consequence, traditional PSMT performs poorly on long distance reordering.

This has brought on a lot of focus on long distance reordering in PSMT, whereas local reordering is often thought to be handled sufficiently by phrases (e.g. Li et al., 2007). However, we believe that short distance reordering can also benefit from linguistic information. This is for example backed by examples like (10), where the reordering is the shortest possible (between two neighboring words), but it relies on non-local information. If a rule states that an NP should change places with a VP with high probability under certain circumstances, then this information should support a PSMT system no matter how many words these constituents span.

3.2.2 Pre-translation reordering as a solution

The problems with reordering in traditional PSMT have been frequently mentioned. In this section, we will describe some of the solutions that have been suggested in previous work.

Much work has been done in this area such as the hierarchical phrases approach (Chiang, 2005; Chiang, 2007) described in section 2.3. As we have chosen to operate within the framework of *pre-translation reordering*, previous work in this area will make up this section.

The pre-translation reordering approach reorders the words of the source sentence prior to translation. The reordering is meant to make the source sentence word order assimilate that of the target language. As described by (Zwarts & Dras, 2007), this approach offers two advantages; 1) better translation word order is achieved, and 2) the strengths of PSMT are better utilized, since non-local phenomena are made local.

In works such as (Xia & McCord, 2004; Collins et al., 2005; Wang et al., 2007; Habash, 2007b), the reordering decisions are done *deterministically*, i.e. a single reordering is presented to the PSMT system. This strategy places the decisions outside the PSMT system by learning to translate from a reordered source language. Other studies (Crego & Mariño, 2007; Zhang et al., 2007b; Li et al., 2007) are more in the spirit of PSMT, in that multiple reorderings are presented to the PSMT system as (possibly weighted) options that are allowed to vote in the log-linear SMT system along side the other parameters. In the following, we will describe these two approaches separately.

Deterministic approaches

The deterministic approaches do not integrate the reordering in the PSMT system; instead they place it outside the system by first reordering the source language, and then using a PSMT system that is trained on reordered source language and target language.

Collins et al. (2005) use six manually created rules for German-English translation. The source sentences are parsed, and the reorderings are done based on a syntax tree. With a system trained on $\sim$15M parallel words in 750K sentences of Europarl data (Koehn, 2005), they improve the BLEU

score from 25.2% to 26.8% on 2000 test sentences.

Wang et al. (2007) replicate the above experiment for Chinese-English with a larger set of rules. The system is trained on 637K sentences of parallel news data. They achieve an improvement in BLEU from 28.52% to 30.86%.

Xia & McCord (2004) use automatically extracted rules for English-French translation. The rules are learned from dependency-parsed parallel sentences. The system is trained on 90M parallel words of the Canadian Hansard corpus, and their approach leads to a decrease in BLEU from 18.7% to 18.5% on 3971 test sentences. On 500 out-of-domain sentences, they however get an improvement from 19.6% to 21.5%.

Habash (2007b) also automatically learns rules but for Arabic-English translation. The rules are learned from only source side dependency-parsed parallel sentences. The system is trained on ∼4M parallel words in 131K of news data. This does not lead to a consistent increase over different test sets.

Non-deterministic approaches

Where the deterministic approach only provides the decoder with one possible source word order as input, the non-deterministic supplies multiple possible reorderings. Usually, these comprise both the original ordering and other rule-predicted reorderings.

Crego & Mariño (2007) operate within Ngram-based SMT, which is comparable to PSMT. They make use of syntactic structure on the source side to automatically learn rules, and with these rules, they reorder the input into a word lattice. Since the paths are not weighted, the lattice merely functions as a linguistically motivated expansion of the monotone search space. The decoder is not given reason to trust one path (reordering) over another. The system is trained on a small corpus of ∼300K parallel Chinese-English words in 46K sentences. Compared to allowing no reordering (monotone decoding), they achieve an improvement in BLEU score from 39.88% to 45.45% on 500 test sentences. In addition, this is an improvement over a system using rules based on POS only that gets 42.47%.

Zhang et al. (2007b) assign weights to the paths of their input word lattice. They automatically learn rules from POS and syntactic chunks, and

train the system with both original and reordered source word order. They also use a restricted data set of ∼400K parallel Chinese-English words in 43K sentences. Their results are somewhat confusing. Their approach outperforms their previous approach (Zhang et al., 2007a), which used a different scoring for reordering and did not train on reordered data with BLEU scores of 59.0% and 60.3%. It, however, does not out-perform a standard PSMT system, which gets 62.4%. What is interesting is that their previous approach outperformed what seems to be the same PSMT system, but on different data (Zhang et al., 2007a). As they themselves point out, a problem in their approach might be that their reordering approach is not fully integrated with PSMT.

Li et al. (2007) use weighted n-best lists as input for the decoder. They use rules automatically learned from a syntactic parse. These rules allow children of a tree node to swap place. Their system translates from Chinese to English, but they do not reveal the size of training or test data. They improve over a standard PSMT system from 29.22% to 30.76% in BLEU.

3.3 Motivation: Problems with previously proposed solutions

Even though several of the approaches described in the preceding section do not explicitly assign probabilities to the source sentence word orders, they do so implicitly. The deterministic approaches assign 100% probability to a given source word order, since it is the only one provided. An unweighted non-deterministic approach assigns equal probability to all source word orders by not distinguishing between them. This can be compared to the weighted non-deterministic approach, which assigns individual, experience-based probabilities to each source word order.

The deterministic approach is problematic in that it makes hard decisions about word order. This can be problematic in a statistical framework. As mentioned by (Al-Onaizan & Papineni, 2006), these deterministic choices are beyond the scope of optimization and cannot be undone by the decoder. First of all, this can make it impossible to make up for bad information in later translation steps. Second, as we showed at the beginning of the chapter, word selection and reordering often depend on each other. In

the deterministic approach, this dynamic relationship does not exist, since reordering has its say first, and then the decoder has to make the best of things based on this.

These concerns convince us that the **non-deterministic approach is superior to the deterministic** when operating within an SMT framework. Our approach will therefore lie within the non-deterministic approaches.

The result of not weighting the word orders is in essence that each is assigned equal probability, even the unordered sentence. This is suboptimal exploitation of the rules, since all the word orders are highly unlikely to be equally probable based on the experience the rules were learned from. If we have seen a reordering take place in a certain context 99% of the times that we have encountered it, then it is misleading to say that there is a 50% chance of the reordering not taking place.

This leads us to believe that **a weighted approach is much more powerful and helpful than an unweighted**, and our approach will therefore use probabilistically weighted rule. We thereby take into consideration the confidence of a rule based on the learning experience.

A general concern with all the pre-translation reordering approaches described in the previous section is that the probability they assign — directly or indirectly — gets assigned to the source side order. We will call this source order (SO) scoring.

The information in a reordering rule concerns a difference between source and target word order. Therefore, when reordering the source sentence to adapt target language word order, the idea is that this word order will transfer to the translation. This is, however, not modelled by previous approaches. Instead, they model the task of producing a source sentence with target language word order. Whether this word order is kept in the final translation or changed to something completely different, is beyond their concern.

The fact is that this word order can indeed change in translation. Even if additional phrase external reorderings are prohibited, it is possible that a phrase internal reordering will change the word order. This means that the phrase internal reordering brings on a discrepancy between source with target word order and target word order in translation, since each source word can make a local move of up to the max length of a phrase to each

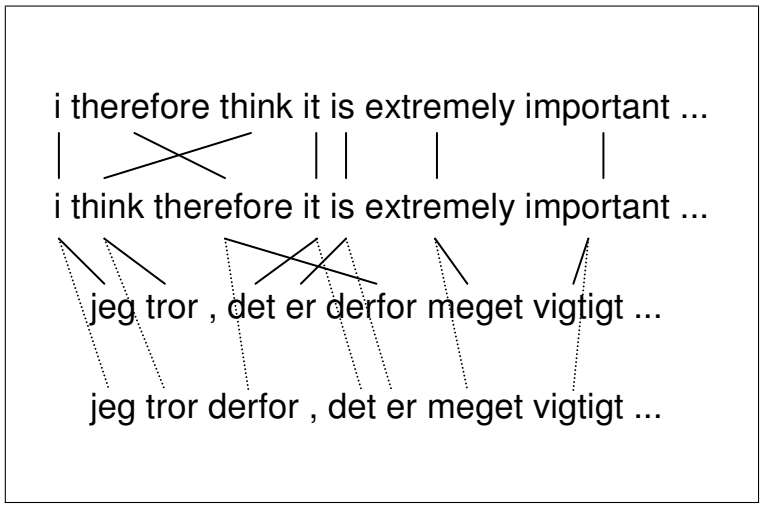


Figure 3.1: Example illustrating the problems with previous approaches.

side. Assigning a probability to a source sentence with target word order based on the reordering rules may therefore prove wrong, since the translation can end up with a very different word order that the rules would actually assign a much different probability.

We illustrated this point earlier in figure 1.1 on page 4. Another example is given in figure 3.1. It contains an excerpt of an English sentence translated by the SO scoring system used on the experiments in chapter 6. First, we have the source sentence, followed by the reordered source sentence. This reordering has been scored as successful by the source side scoring mechanism. The phrase table, however, contains a phrase pair *'therefore it is → det er derfor'*, which often is a good reordering. This results in *therefore* moving additionally two positions to the right and a poor translation, without the scoring mechanism knowing about it. The translation is assigned a score for making a good reordering without actually making this reordering. Had the translation kept the rule-proposed word order, the translation would have been much better, as illustrated by the final sentence.

In short, there is a discrepancy between the SO scoring model and the task it is meant to model. This may mislead the translation, since the probability assigned to a reordering is not necessarily consistent with the learning experience of the rules. We therefore believe that an approach that scores on the target side avoids these problems and thereby that **a target**

side scoring approach is superior to one that scores on the source side.

An additional issue in connection with source side reordering is the word order of the training data. The deterministic approaches reorder the training data, so the system translates from reordered source language to target language.

Doing this leads to a less complex relation between the source and the target sentence, which may provide a better basis for learning phrase pairs. If there is less reordering between a sentence pair, it should be possible to learn longer and more phrases. To our knowledge, this connection has not been examined.

It also provides the system with better odds, since it will be translating from exactly the language it has been trained on. Otherwise, it uses a source-target system to do translation between reordered source and target.

This approach is not directly adaptable with a probabilistic, non-deterministic environment, since there does not exist a single source side word order. Instead, there are multiple orderings, which are not equally probable.

Zhang et al. (2007b) try to circumvent these problems by aligning the same data twice; once without modifications and once with reordered source language. A phrase table is then extracted from the combined data. The source language is reordered based on the alignment from the unmodified data. Syntactic chunks on the source side are rearranged, so they get the same order as the target side according to the alignment. They report an insignificant BLEU increase of 0.6% from this modification.

This approach may diminish the divergence between the training language and the language to be translated, but it still exists. The gap can only be mended by creating both data sets in the same way, i.e. using the rules, as was the case with the deterministic approaches. It is necessary to find a way of extracting phrase pairs from a word lattice that is aligned to a string, and perhaps these phrase pairs ought to be weighted by the probability of the given word order.

This is a very interesting subject, but we will restrict ourselves from going further into phrase extraction based on reordered source language for probabilistic, non-deterministic approaches in this thesis.

In summation, we believe that the optimal incorporation of pre-

translation reordering in PSMT is a probabilistic, non-deterministic approach. However, the previous approach of scoring the source side word order is problematic. Instead, we propose scoring the target word order as a superior means for controlling reordering.

We will now turn to the word alignments need for extracting the reordering rules.

Chapter 4

Improving word alignment

Word alignments over parallel corpora have become an essential supporting technology to a variety of natural language processing (NLP) applications, most prominent among which is statistical machine translation (SMT). Although phrase-based approaches to SMT tend to be robust to word-alignment errors (Lopez & Resnik, 2006), improving word-alignment may still be meaningful for many other NLP research areas that is more sensitive to alignment quality, e.g., projection of information across parallel corpora (Yarowsky et al., 2001), or in our case, as basis for learning reordering rules.

In this chapter[1], we present a novel approach to using and combining multiple preprocessing (tokenization) schemes to improve word alignment. The focus will be on English-Arabic. The produced alignment will in the next chapter be used as a resource for reordering rule extraction. The idea is to examine the effect that the choice of word alignment method has on the rules learned. In our word alignment approach, the text to align is tokenized before statistical alignment and is then remapped to its original form afterward. Multiple tokenizations yield multiple *remappings* (remapped alignments), which are then combined using supervised machine learning. The intuition here is similar to the combination of different preprocessing schemes for a morphologically rich language as part of SMT (Sadat & Habash, 2006) except that the focus is on improving the alignment quality.

[1] The work presented in this chapter is joint work with Nizar Habash. This work has previously been published in (Elming & Habash, 2007) and (Elming et al., to appear).

In the following two sections, we present related work and Arabic preprocessing schemes. Section 4.3 and 4.4 present our approach to alignment preprocessing and combination, respectively. Alignment results are presented in Section 4.5.

4.1 Related work

Recently, several successful attempts have been made at using supervised machine learning for word alignment (Liu et al., 2005; Taskar et al., 2005; Moore, 2005; Ittycheriah & Roukos, 2005; Fraser & Marcu, 2006; Cherry & Lin, 2006). This approach often makes for faster alignment and is easier to add new features to, compared to generative models. With the exception of (Moore, 2005) and (Fraser & Marcu, 2006), the abovementioned publications do not entirely discard the generative models. Instead, they integrate IBM model predictions as features. We extend on this approach by including IBM alignment information based on multiple preprocessing schemes in the alignment process.

In other related work, (Tillmann et al., 1997) use several preprocessing strategies on both source and target language to make them more alike with regards to sentence length and word order. (Lee, 2004) only changes the word segmentation of the morphologically complex language (Arabic) to induce morphological and syntactic symmetry between the parallel sentences.

We differ from previous work by including alignment information based on multiple preprocessing schemes in the alignment process. We do not decide on a certain scheme to make source and target sentences more symmetrical with regards to the number of tokens and their content. Instead, it is left to the alignment algorithm to decide under which circumstances to prefer alignment information based on one preprocessing scheme over information based on another scheme.

The intuition behind using different preprocessing schemes for word alignment is a simple extension of the same intuition for preprocessing parallel data for SMT. Namely, that reduction of word sparsity often improves translation quality (and in our case alignment quality). This reduction can be achieved by increasing training data or via morphologically

Preprocessing Scheme		Example	
Name	**Definition**	**Arabic Script**	**Transliteration**
$NONE$	natural text	وسيكتبها!	*wsyktbhA!*
AR	simple tokenization	وسيكتبها !	*wsyktbhA !*
$D1$	decliticize CONJ	و+ سيكتبها !	*w+ syktbhA !*
$D2$	decliticize CONJ, PART	و+ س+ يكتبها !	*w+ s+ yktbhA !*
TB	Arabic Treebank tokenization	و+ سيكتب +ها !	*w+ syktb +hA !*
$D3$	decliticize all clitics	و+ س+ يكتب +ها !	*w+ s+ yktb +hA !*

Table 4.1: Arabic preprocessing scheme variants for وسيكتبها! 'and he will write it!'

driven preprocessing (Goldwater & McClosky, 2005). Recent publications on the effect of morphology on SMT quality focused on morphologically rich languages such as German (Nießen & Ney, 2004); Spanish, Catalan, and Serbian (Popović & Ney, 2004); Czech (Goldwater & McClosky, 2005); and Arabic (Lee, 2004; Habash & Sadat, 2006). They all studied the effects of various kinds of tokenization, lemmatization and POS tagging and show a positive effect on SMT quality. However, to our knowledge, no study tried to tease out the effect of tokenization on word alignment. Sadat & Habash (2006) investigated the effect of combining multiple preprocessing schemes on MT quality in a PSMT system. In this chapter, we focus on alignment improvement independent of SMT.

4.2 Arabic preprocessing schemes

Arabic is a morphologically complex language with a large set of morphological features. As such, the set of possible preprocessing schemes is rather large (Habash & Sadat, 2006). We follow the use of the terms *preprocessing scheme* and *preprocessing technique* as used by (Habash & Sadat, 2006). We focus here on a subset of schemes pertaining to Arabic attachable clitics. Arabic has a set of attachable clitics to be distinguished from inflectional features such as gender, number, person and voice. These clitics are written attached to the word and thus increase its ambiguity. We can classify three degrees of cliticization that are applicable in a strict order to a word base:

```
[CONJ+
    [PART+
        [Al+ BASE +PRON]]]
```

At the deepest level, the BASE can have a definite article +ال (*Al*+[2] 'the') or a member of the class of pronominal clitics, +PRON, (e.g., +ها *hA*+ 'her/it/its'). Pronominal enclitics can attach to nouns (as possessives) or verbs and prepositions (as objects). Next comes the class of particles (PART+), (e.g., +ل *l*+ 'to/for' or +سـ *s*+ 'will/future'). Most shallow is the class of conjunctions (CONJ+), (+و *w*+ 'and' and +ف *f*+ 'then').

We use the following five schemes: AR, $D1$, $D2$, $D3$ and TB. Definitions and contrastive examples of these schemes are presented in Table 4.1. To create these schemes, we use MADA (the Morphological Analysis and Disambiguation for Arabic), an off-the-shelf resource for Arabic morphological disambiguation (Habash & Rambow, 2005), and TOKAN, a general Arabic tokenizer (Habash, 2007a).

4.3 Preprocessing schemes for alignment

4.3.1 Giza++ alignments

The basic alignments used as baselines and by the combiner in this work are created with the Giza++ statistical aligner (Och & Ney, 2003). Giza++ is a somewhat extended implementation of the IBM models (Brown et al., 1993) that was described in section 2.2.1. To sum up, the IBM models 1–5 use increasingly sophisticated modeling to achieve better alignments based on non-linguistic, statistical information about word occurrences in language parallel sentences.

A limitation in the IBM models is that they create asymmetrical alignments, i.e., they only allow one-to-many linking from source to target. In order to make the alignments symmetrical, heuristics that combine two alignments trained in opposite directions are often applied. By combining the one-to-many and many-to-one alignments, it is possible to obtain a symmetrical many-to-many alignment. For our baseline alignment, we

[2] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter transliteration scheme (Habash et al., 2007).

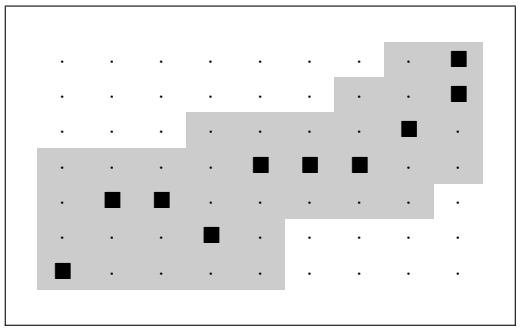


Table 4.2: Word alignment illustrating the pruned search space for the combiner.

chose the GDF symmetrization heuristic described in section 2.2.2. This heuristic adds links to the intersection of two asymmetrical statistical alignments in an attempt to assign every word a link.

4.3.2 Alignment remapping

Using a preprocessing scheme for word alignment breaks the process of applying Giza++ on some parallel text into three steps: preprocessing, word alignment and remapping. In preprocessing, the words to align are tokenized into smaller units. Then, they are passed along to Giza++ for alignment (default settings). Finally, the Giza++ alignments are mapped back (remapped) to the original word form before preprocessing. In this work, *all* words are AR tokens because our hand-aligned training and test data are in this scheme (Section 4.5.1). However the alignment is done using different schemes. For instance, take the first word in Table 4.1, *wsyktbhA*; if the $D3$ preprocessing scheme is applied to it before alignment, it is turned into four tokens (*w+ s+ yktb +hA*). Giza++ will link these tokens to different words on the English side (e.g., 'and he will write it'). In the remapping step, the union of these links is assigned to the original word *wsyktbhA*. We refer to such alignments as remappings.

4.4 Alignment combination

After creating the multiple remappings, we pass them as features into an alignment combiner. The combiner is also given a variety of additional

features, which we discuss later in this section. The combiner is a binary classifier that determines for each source-target word pair whether they are linked or not. Because of the large amount of data used, we use a simplifying heuristic that allows us to minimize the number of source-target pairs used in training. Only links evidenced by at least one of the initial alignments and their immediate neighbors are included. This is exemplified by the matrix in table 4.2 which illustrates an alignment (the black squares) and the search space (the gray area) attained by expanding to all immediate surrounding neighbors of all links, i.e., all points bordering on the side or corner of a link. This provides the bottom left link with only 3 neighbors, while a centered link has 8 neighbors. All other links (the white area) are considered non-existent. This choice removes a large portion of the search space, but at the expense of raising the lower performance boundary for the system. On the development data set, 78.6% of the search space is removed at the expense of removing 2.2% of the correct links. This means that the lowest possible error rate is 2.2%.

The combiner we use here is implemented using a rule-based classifier, Ripper (Cohen, 1995; Cohen, 1996). Ripper is a rule induction algorithm that builds on the ideas of decision trees. First, a rule is grown by continually adding the feature the reduces entropy most. This results in a largely overfitting set of rule. The rules are therefore optimized against a held-out validation set by pruning away as many rule conditions as possible without hurting performance as measured by a loss function.

The reasons we use Ripper as opposed to other machine learning approaches are: (a) Ripper produces human readable rules that allow better understanding of the kind of decisions being made; and (b) Ripper is faster than the other machine learning approaches we examined for the very large amount of training data we used.[3] The combiner is trained using supervised data (human annotated alignments), which we discuss in Section 4.5.1.

In the rest of this section we describe the different machine learning features given to the combiner. We break the combination features in two types: word/sentence features and remapping features.

[3] In a small pilot experiment, Ripper used 4 hours of training time, and TinySVM used 4 days (http://chasen.org/∼taku/software/TinySVM/).

Word/sentence features

- **Word Form (WW)**: The source and target word forms.
- **POS (WP)**: The source and target part-of-speech tags. For Arabic, we use the Bies POS tagset (Maamouri et al., 2004) as output by MADA. For English, we use MXPOST (Ratnaparkhi, 1996) trained on the Penn Treebank (Marcus et al., 1994).
- **Location (WL)**: The source and target *relative* sentence position (the ratio of absolute position to sentence length). We also include the difference between the source and the target relative position.
- **Frequency (WF)**: The source and target word frequency computed as the number of occurrences of the word form in training data. We also use the ratio of source to target frequency.
- **Similarity (WS)**: This feature is motivated by the fact that proper nouns in different languages often resemble each other, e.g. صدام حسين *SdAm Hsyn* and 'saddam hussein'. We use the equivalence classes proposed by (Freeman et al., 2006) to normalize Arabic and English word forms (e.g. the former example becomes 'sdam hsyn' and 'sadam husyn'). Then, we employ the longest common subsequence as a similarity measure. This produces the longest (not necessarily contiguous) sequence that the two compared sequences have in common. The similarity score is calculated as the intersection (i.e. the number of characters in the longest common subsequence) over the union (i.e. intersection + non-matching characters) (the former example gets a similarity score of 8/(8+2) = 0.8).

Remapping features

- **Link (RL)**: for each possible source-target link, we include (a) a binary value indicating whether the link exists according to each remapping; (b) a cumulative sum of the remappings supporting this link; and (c) co-occurrence information for this link. This last value is calculated for each source-target word pair as a weighted average of the product of the relative frequency of co-occurrence in both directions for each

remapping. The weight assigned to each remapping is computed empirically.[4] Only the binary link information provides a different value for each remapping. The other two give one combined value based on all included remappings.

- **Neighbor (RN)**: The same information as Link, but for each of the (three to eight) immediate neighbors of the current possible link individually. These features inform the current possible link about whether its surrounding points are likely to be links. This is motivated by the fact that alignments tend towards making up a diagonal line of adjacent points in the alignment matrix.

- **Cross (RC)**: These include (a) the number of source words linked to the current target word; (b) the number of target words linked to the current source word; (c) the sum of all links to either the current source word or the current target word; (d) the ratio of the co-occurrence mass between the current target word and the current source word to the total mass between all target words and the current source word; (e) same ratio as in (d) but in the other direction; and (f) the ratio of the total co-occurrence mass assigned to either the current source word or the current target word to the co-occurrence mass between the current target word and the current source word. With these features, we obtain a relation to the rest of the sentence. This provides information on whether there are better ways of linking the current source and target word respectively.

4.5 Evaluation

A basic assumption for our investigation is that statistical word alignments based on different preprocessing schemes will lead to different systematically detectable advantages. A machine learning algorithm should as a consequence profit from the information made available by doing word alignment based on several different preprocessing schemes as opposed to a single scheme. In order to test this hypothesis, we conduct the following

[4] We use the AER on the development data normalized so all weights sum to one. See Section 4.5.3.

four experiments with the goal of assessing:

1. the contribution of alignment remapping (Section 4.5.2),
2. the contribution of combination features for a single alignment, i.e., independent of the combination task (Section 4.5.3),
3. the contribution of the individual features (Section 4.5.4), and
4. the best performing combination of alignment remappings (Section 4.5.5).

All of these experiments are done using a development set. We then pick our best performing system and use it on a blind test set in Section 4.5.6. We also present an analysis of the rules we learn in Section 5.5 and an error analysis of our best development system in Section 4.5.8. Next, we discuss the experimental setup and metrics used in all of these experiments.

4.5.1 Experimental data and metrics

Data sets

The gold standard alignments we use here are part of the IBM Arabic-English aligned corpus (IBMAC) (Ittycheriah & Roukos, 2005). Of its total 13.9K sentence pairs, we only use 8.8K sentences because the rest of the corpus uses different normalizations for numerals that make the two sets incompatible. We break this data into 6.6K sentences for training and 2.2K sentences for development by letting every fourth line go to the development set. As for test data, we use the IBMAC's test set (NIST MTEval 2003 – 663 sentences with four references, all Arabic-English aligned). Experiments in Sections 4.5.3, 4.5.4, and 4.5.5 used only 2.2K of the gold alignment training data (not the same as the development set) to minimize computation time. As for our test experiment (Section 4.5.6), we use our best system with all of the available data (8.8K).

To get initial Giza++ alignments, we use an Arabic-English parallel corpus of about four million words of newswire (LDC-NEWS) for training data together with the annotated set. The parallel text includes Arabic

News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18).[5]

Since the IBMAC and LDC-NEWS have much overlap, we take care to remove duplications in LDC-NEWS to avoid biasing our experiments. The additional data (LDC-NEWS minus IBMAC) was prepared to match the preprocessing scheme used in IBMAC (AR with some additional character normalizations). We match the preprocessing and normalizations on our additional data to that of IBMAC's Arabic and English preprocessing (Ittycheriah & Roukos, 2005).

Metrics

The standard evaluation metric within word alignment is the Alignment Error Rate (AER) (Och & Ney, 2000), which requires gold alignments that are marked as 'sure' or 'probable'. Since the IBMAC gold alignments we use are not marked as such, AER reduces to 1 - F-score (Ittycheriah & Roukos, 2005):

$$Pr = \frac{|A \cap S|}{|A|} \quad Rc = \frac{|A \cap S|}{|S|} \quad AER = 1 - \frac{2PrRc}{Pr+Rc}$$

where A links are proposed and S links are gold. Following common practice, NULL links are not included in the evaluation (Ayan, 2005; Ittycheriah & Roukos, 2005). In addition to AER, we also use Precision (Pr) and Recall (Rc) in some cases to better compare different systems.

The baseline we measure against in all of the experiments in this section is the GDF symmetrization algorithm discussed earlier in Section 4.3. The AER of this baseline is 24.77 for the development set and 22.99 for the test set.

4.5.2 The contribution of alignment remapping

We experimented with five alignment remappings in two directions: *dir* (Ar-En) and *inv* (En-Ar). Table 4.3 shows the AER associated with each of the ten alignment remappings and the remapping of their corresponding GDF symmetrized alignment (*gdf*). Table 4.3 also contains information on

[5] All of the training data we use is available from the Linguistic Data Consortium (LDC): http://www.ldc.upenn.edu/.

Remapping	**Word count**	*dir*	*inv*	*gdf*
AR	47721	24.67	31.68	24.77
$D1$	50584	23.07	28.16	22.90
$D2$	52900	22.17	25.29	21.63
TB	54507	21.50	23.93	21.04
$D3$	65787	20.76	22.35	20.45

Table 4.3: AER and word count for each Alignment Remapping in both directions and combined using the GDF heuristic.

the word count of the schemes, which corresponds to an English text with 58201 word tokens. The more segmented a preprocessing scheme (i.e., the greater the word count), the lower the AER for either direction and for *gdf* of the corresponding remapping. The order of the schemes from worst to best is AR, $D1$, $D2$, TB and $D3$. INV alignments are always worse than DIR alignments. This indicates the difference between Arabic and English morphology. The more you split up the Arabic words, the easier it becomes to match them to their English correspondences. Even when the Arabic word count exceeds the English with more than 7500 tokens, we still get an improvement. The results reveal that the remapping strategy in itself is an interesting approach to alignment. When interested in word aligned text in a specific preprocessing scheme, it might be worth doing word alignment in a different scheme followed by a remapping step. The best result we obtained through remapping is that of $D3_{gdf}$ which had a 20.45% AER (17.4% relative decrease from the baseline).

4.5.3 The contribution of combination features

This experiment is conducted to specify the order for combining the alignment remappings when finding the overall best system (see Section 4.5.5). For each of the basic ten (non *gdf*) alignment remappings, we trained a version of the combiner that uses all the relevant features but has access to **only** one alignment at a time.

The results of evaluating on the development data are show in Table 4.4. We see a substantial improvement resulting from applying the alignment

Alignment Remapping	**AER**
AR_{inv}	20.79
$D1_{inv}$	19.30
$D2_{inv}$	17.77
AR_{dir}	17.26
TB_{inv}	16.77
$D1_{dir}$	16.35
TB_{dir}	16.14
$D3_{inv}$	15.83
$D2_{dir}$	15.56
$D3_{dir}$	14.50

Table 4.4: AER for the combination system when Alignment remappings are varied.

combination as a *supervised alignment correction system*. For the 10 alignment remappings the AER ranges from 14.5 to 20.79, giving an average relative improvement of 29.9% (down from 20.76 to 31.68 in columns three and four in Table 4.3). The relative order of all alignments remains the same with this improvement except for TB_{dir} which moves from #2 to #4. In addition to determining the order of combination, the scores in Table 4.4 are also used to weigh the co-occurrence information supplied by each alignment remapping as described in footnote 4 in Section 4.4.

4.5.4 The contribution of individual features

In order to validate the importance of each feature cluster to the alignment algorithm, a two step experiment is conducted. First, each feature cluster is removed individually from the best performing system from the previous experiment (AER($D3_{dir}$) = 14.50). The increase in AER indicates the importance of this feature cluster. Secondly, the features are added cumulatively in the order of importance to determine the best combination of features.

The results listed in Table 4.5 show that all of the features help the alignment algorithm, and the best combination of features includes all of them. Not surprisingly, the alignment features are more important than the word

Feature Cluster	Remove	Add Cumulative
AC: Alignment Cross Link	16.32	19.97
AN: Alignment Neighbor Link	16.14	17.29
AL: Alignment Basic Link	16.02	17.07
WF: Word Frequency	15.28	15.49
WP: Word Position	15.01	14.82
WW: Word Form	14.97	14.75
WL: Word Location	14.78	14.77
WS: Word Similarity	14.77	14.50

Table 4.5: The effect of varying feature clusters in the combination system.

features.

4.5.5 Alignment combination experiments

To determine the best subset of alignment remappings to combine, we ordered the remappings given their AER performance when used individually in the combination system (Section 4.5.3). This was done by forward selection. Starting with the best performer ($D3_{dir}$), we continue adding alignments in the order of their performance so long the combination's AER score is decreased. Our combination results are listed in Table 4.6. The best alignment combination used alignments from four different schemes which confirms our intuition that such combination is useful.

We further trained our best combination on all of the training data (6.6K sentences) as opposed to only 2.2K training sentences (see Section 4.5.1). The best combination performance improves slightly to 12.24 from 12.69.

4.5.6 Test set evaluation

We ran our best system trained on all of the IBMAC data (training & development), on all of the unseen IBMAC test set. The results are shown in Table 4.7 comparing training on all seen data (training and development) to just using the training data. The development set shows a relative improvement of 50.6% (24.77 to 12.24). On the test data, we also achieve a sub-

Alignment Remapping combination	**AER**
$D3_{dir}$	**14.50**
$D3_{dir}D2_{dir}$	**14.12**
$D3_{dir}D2_{dir}D3_{inv}$	**12.81**
$D3_{dir}D2_{dir}D3_{inv}TB_{dir}$	13.05
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}$	**12.75**
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}TB_{inv}$	12.78
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}AR_{dir}$	12.84
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}D2_{inv}$	12.76
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}D1_{inv}$	12.93
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}AR_{inv}$	**12.69**

Table 4.6: Determining the best combination of alignment remappings

Data	**Development**	**Test**
Baseline	24.77 (76.45 / 74.04)	22.99 (72.39 / 82.25)
TRAIN (6.6K)	**12.24** (88.43 / 87.11)	14.31 (80.17 / 92.02)
ALL (8.8K)	—	**14.19** (80.46 / 91.93)

Table 4.7: Development vs. Test results: AER (Precision / Recall)

stantial relative improvement of 38.3% when using all training data (22.99 to 14.19).

On the test data, the initial search space reduction heuristic behaves much as on the development and training data. The search space is reduced by around 80% and in the processes only 1.4% of the correct links are removed. In other words, the lower boundary for the system is an AER of 1.4.

The test baseline is lower than the development baseline, yet the best AER on test is higher than development. The precision and recall measures give additional insights into this issue. The test baseline is much higher in terms of its recall compared to the development baseline; however its precision is lower. This trade-off pattern is preserved in our best systems.

This large difference between precision and recall also corresponds to a disproportionate number of links in the test baseline compared to the the test reference: test alignment links are 14% more than test reference, compared to development alignment links, which are 3% *less* than their reference. One possible explanation of this difference between development and test is that the test data in fact contains four replicas in the Arabic with different English translations (see Section 4.5.1). Since all of the initial alignments were done jointly, the performance on this subset may be biased, especially in terms of recall. Nonetheless, our approach improves both precision and recall for both development and test. The last experiment, using all of the data for training, gives a small boost to the test AER score, but the improvement seems to be specifically in terms of an increase in precision, together with a tiny decrease in recall.

(Ittycheriah & Roukos, 2005) used only the top 50 sentences in IBMAC test data. Our best AER result on their test set is 14.02 (baseline is 22.48) which is higher than their reported result (12.2 with 20.5 baseline (Arabic-to-English GIZA++)). The two results are not strictly comparable because: (a) (Ittycheriah & Roukos, 2005) used additional gold aligned data that was not released and (b) they used an additional 500K sentences from the LDC UN corpus for Giza training that was created by adapting to the source side of the test set – the details of such adaptation were not provided and thus it was not clear how to replicate to compare fairly. Clearly this additional data is helpful since even their baseline is higher than ours.[6]

4.5.7 Alignment rule analysis

The rules provided by Ripper have the advantage of giving us an insight into the choices made by the classifier. In this section, we look closely at three rules selected from our best performing system.

First, the number one rule learned by Ripper is also the simplest and most commonly applied. It has a precision of 97.0% and a recall of 67.3%. The rule simply states that a link should be assigned if both $D3_{dir}$ and $D3_{inv}$ contain this link. In other words, the backbone of the combination alignment is the intersection of both directions of the $D3$ remappings.

[6] Abraham Ittycheriah, personal communication.

Second, the number two rule learned by Ripper is more complex and thus less general. It has a precision of 99.0% and a recall of 2.4% (of what is left after rule number one applies). The rule contains the following conditions:

1. RC(a) $D2_{dir}$ = 1
2. RL(a) AR_{inv} = 1
3. RC(f) $>= 0.15$
4. WS $>= 0.44$

The first condition states that according to the $D2_{dir}$ remapping, the Arabic word should only link to this English word. In fact, due to the unidirectionality of GIZA++ alignments, this means that the two words should only align to each other (in the $D2_{dir}$ remapping). The second condition says that the AR_{inv} remapping should contain the link. The third condition requires that this link carry at least 15% of the combined lexical probability of the source and target words being linked to any word. Finally, the fourth condition states that the two word forms should at least to a certain degree be similar. The majority of cases handled by this rule are multi-word expressions (in Arabic, English or both) where the words being linked by the rule are similar to some degree but the links were missed by $D3_{dir}$ or $D3_{inv}$ (thus, rule number one did not apply).

The last rule we examine here applies as the 23rd rule of the set. It has a precision of 89.8% and a recall of 0.9% (of remaining links). The rule contains the following conditions:

1. WP(Arabic) = NN
2. WP(English) = DT
3. RN(right) $D3_{dir}$ = 1
4. WL(difference) $<= 0.05$
5. RC(c) $<= 9$

The first two conditions state that the Arabic word is a noun, and the English is a determiner. The third states that the right neighbor should be linked according to the $D3_{dir}$ remapping. In other words, this reveals that the Arabic word should be linked to the following English word as well according to $D3_{dir}$. The difference in relative sentence position should be small, i.e., the words should appear about the same place in the sentence. And finally, the two words should not have a lot of other linking options in the available remappings. In other words, an Arabic noun should link to an English determiner, if the Arabic noun is also linked to the following English word (quite possibly a noun). This rule handles the fact that the determiner is often a part of the Arabic word, which is not the case in English. Only the $D3$ tokenization scheme separates the *Al+* determiner in Arabic.

4.5.8 Error analysis

We conducted a detailed error analysis on 50 sentences from our development set's baseline and best system. The sample included 1011 Arabic words and 1293 English words. We found 465 erroneous alignments (including null alignments) in the baseline and 218 errors in our best system. We classified errors as follows. *Closed-class*[7] errors involve the misalignment of a closed-class word in Arabic or English. *Open-class* errors involve open-class words such as nouns and verbs. *Numeral* and *Punctuation* errors involve numbers and punctuation misalignments, respectively. Finally, *Compositional* errors are complex errors involving non-compositional expressions (as in the idiom *half-brother* mapping to the Arabic اخ غير شقيق *Ax gyr šqyq*, lit. *brother not full-brother*) or compositional translation divergences (as in *aggravate* mapping to the Arabic زاد تفاقما *zAd tfAqmA*, lit. *increase aggravation*) (Dorr et al., 2002). Open-class and closed-class errors are strictly defined here to not involve compositional errors. We also computed *gold* errors in our best system; these are cases inconsistent with the alignment guidelines as explained in (Ittycheriah & Roukos, 2005).

Table 4.8 presents the results of the error analysis. The first column lists the different error classes. The second and third columns list the frequency

[7] As opposed to an open-class, a closed-class is a relatively small group of words that is usually not extended by new words. Determiners, prepositions and pronouns are examples of closed word classes.

Error Type	**Baseline Frequency**	**Best System Frequency**	**Error Reduction**	**Best System Gold Errors**
Closed	236 (51%)	117 (54%)	50%	18 (15%)
Open	117 (25%)	33 (15%)	72%	0 (0%)
Comp	89 (19%)	50 (23%)	44%	19 (38%)
Num	13 (3%)	9 (4%)	31%	0 (0%)
Punc	10 (2%)	9 (4%)	10%	2 (22%)
Total	465	218	53%	39 (18%)

Table 4.8: A categorization of alignment errors found in error analysis of baseline and best performing system.

of errors in the baseline and best system, respectively. The percentages in parentheses indicate the ratio of the error type in that column. The fourth column specifies the error reduction in our best system from the baseline. Overall, we reduce the errors by over 50%. The largest reduction is in the open-class errors followed by closed-class errors. The relative distribution of errors is similar in baseline and best system except that open-class and compositional errors exchange ranks: open-class errors are second in the baseline but third in our best system. The last column lists the frequency of gold errors. The percentages in parentheses are ratios against the corresponding best-system frequencies. The gold errors comprised 18% of all of the errors in our best system. They are generally split between closed-class and compositional errors.

The errors in our best system are consistent with previous studies where a majority of errors is associated with closed-class words (especially determiners such as *the*). Closed-class errors can be attributed to their high frequency as opposed to the open class errors which are more a result of their low frequency or out-of-vocabulary status. Compositionality errors are complex and seem to result from there being no clear definition on what is compositional on one hand and from a lack of a multi-word alignment model in our system. The need for a way to enforce a well-formed multi-word alignment (or phrasal constructs in general) is also responsible for many of the closed-class errors. Such a multi-word alignment model would

be an interesting extension of this research since it explores the meaning of an *alignment token* at different levels above and below the *word*. Perhaps, one could use a phrase chunker or even a parser to add constraints on either alignment or combination steps (Cherry & Lin, 2006).

Punctuation errors are perhaps a result of lower use of punctuation in Arabic as opposed to English, thus, there is a lot of sparsity in the training data. Number errors are a result of neutralizing all numerals in our system, i.e., merging them all to a single number category. If the word form of the numbers had not been a neutralized expression, the word similarity measure would probably have made the system capable of handling numbers better.

4.6 Conclusion

In this chapter, we have presented an approach for using and combining multiple alignments created using different preprocessing schemes. Our results show that the remapping strategy improves alignment correctness by itself. We also show that the combination of multiple remappings improves word alignment measurably over a commonly used state-of-the-art baseline. We obtain a relative reduction of alignment error rate of about 38% on a blind test set. In the next chapter, we will turn to the learning of reordering rules from word alignments.

Chapter 5

Learning reordering rules

One basic motivation for doing statistical machine translation is the time consuming and expensive nature of developing rule-based machine translation systems. For the same reasons, it is also compelling to automatically learn reordering rules instead of constructing them manually, if these are to be integrated in an SMT system. This way, the basic motivation is kept intact.

Where most approaches for learning reordering rules automatically result in a very large amount of rules, our focus is primarily on the most general and frequent of rules. This is inspired by the experiments conducted by e.g. Collins et al. (2005) and Wang et al.(2007), who achieve improved translation based on a few manually created, general rules.

In this chapter, we will describe how reordering rules have been learned in previous work, and how we have chosen to handle it in the present experiments. Usually, word alignments play an important role in this task. We will also examine the effect that the choice of alignment method, domain and the size of the training data has on the rules learned. In this chapter, evaluation is performed as a manual analysis of the rules. Experimental results are reported in the next chapter together with the reordering approach.

5.1 Related work

A lot of work has been done in automatically learning reordering rule both in and outside of SMT. We will here focus on the most directly related work. Much of this work was also described in section 3.2.2, but here we will focus on their rule extraction.

Xia & McCord (2004) use dependency parses on both source and target side. They extract rewrite patterns for all nodes in their parse trees, which means that reordering is restricted to children of a single node (*sister node reordering*). Lexicalized rules annotated with word forms are also extracted. Due to the exponential growth of this, they restrict the possible rule length. Still, they get millions of rules. Rules are organized in a hierarchy, where more specific rules overrule more general ones. This means that the rules are heuristically prioritized instead of probabilistically. They are extracted from 90M parallel English-French words, which results in 56K rules after heuristic reduction.

Habash (2007b) also extracts rewrite patterns, but from data that has only been dependency-parsed on the source side. The rules are not lexicalized, and reordering is restricted to sister node reordering, but prioritizing is here done based on probabilities calculated from occurrences in the training data. Rules are extracted from 131K parallel Arabic-English sentences (4.1M/4.4M words), resulting in 71K rules. Examining different sets of statistical alignments, Habash finds that the intersection of statistical alignments in both language directions provides the most consistent rules.

Crego & Mariño (2007) also extract rules based on source side syntactic dependency structures. The reordering is bound to the syntactic structure in that a single node should cover the entire reordering, but it is not restricted to sister node reordering. The rules are not lexicalized, and no prioritizing is executed giving the individual rules no measure of confidence. 46K sentences of parallel Chinese-English data (326K/314K words) is used to extract the rules. The total number of rules is not reported, but it is expectedly high, since 3,264 of the rules apply to the 500 test sentences (3K words).

Zhang et al. (2007a) do not employ a hierarchical structure. Instead, they use a chunker to get information on linguistic phrases. They extract rules

for reordering two neighboring chunk sequences. These sequences are restricted by the phrase extraction algorithm of (Zens et al., 2002). That is, if the corresponding target side word sequences overlap, then this data is discarded. Their rules are, however, not restricted by a hierarchical structure, and they are not lexicalized. In addition, they are not prioritized, but a language model is trained on reordered source to evaluate word orders created by the rules. In (Zhang et al., 2007b), this is supplement by assigning a probability to each rule. Rules are extracted from 18K sentences (486K words), and they do not report how may rules are extracted, but on 1K sentences (22K words) of test data, 3685 of the rules apply.

As opposed to the approaches above, Li et al. (2007) use advanced learning techniques to learn reordering. This is based on a binary syntax tree and restricted to sister node reordering. They train a Maximum Entropy (ME) model to learn whether two sister nodes should exchange places. They include the following features for each of the two sister phrases; leftmost, rightmost, head, and context words and their POS. The context word is the single word bordering on the opposite side of the reordering candidate phrase. The same restriction as with Zhang et al. (2007a) applies here; if target sequences overlap, the data is omitted. The training data size is not revealed, and the number of rules in a ME model must be considered large.

5.2 Positioning within relevant work

In this section, we will discuss the approaches described above and based on this, position the present approach to rule learning within these. Table 5.1 visualizes the contrasts between the present approach and the previous work described.

Zhang et al. (2007a) distinguish themselves from the others in that they do not employ a syntax tree. This may not provide enough information to deal with certain reorderings, as described in section 3.1, English-Danish reordering for example requires subordination information, which can be be obtained from a syntactic structure. We therefore choose to incorporate hierarchical information in our approach.

Only some of the approaches allow lexicalized rules. This is a very strong and important feature in dealing with language variance. For example,

Related work	Hierarchical structure	Lexical information	Non-sister reordering	Probabilistic reordering	Advanced learning	Include crossing phrases
(Xia & McCord, 2004)	+	+	−	−	−	+
(Habash, 2007b)	+	+	−	+	−	+
(Crego & Mariño, 2007)	+	−	+	−	−	+
(Zhang et al., 2007b)	−	−	+	+	−	−
(Li et al., 2007)	+	+	−	+	+	−
Present work	+	+	+	+	+	−

Table 5.1: Schema positioning the present work to previous approaches.

dealing with the fact that only a minority of French adjectives appear in front of the noun, may be hard to handle without lexical information in the rules. In the present work, we include lexical information.

Most of the approaches described above are restricted to reordering sister nodes. Galley et al. (2004) report that in their experiments, this restriction means that the rules are only able to cover 12.1% to 19.4% of the reorderings encountered in different corpora. To exemplify the problem, a common reordering in English-Danish translation has the subject change place with the finite verb. Since the verb is often embedded in a VP containing additional words that should not be moved, such rules cannot be captured by local reordering on tree nodes. On account of this, we choose an approach where the reordering is not restricted be the hierarchical structure.

In most of the approaches, predefined feature patterns are sought out and counted. This may be too restrictive, since there is little flexibility in which features make up a rule, and how important they are for the reordering. Utilizing a more advanced machine learning approach, such as Li et al. (2007) does, makes it possible to learn for a given reordering, which features are important, and which are at all relevant. In their experiments, the ME model outperforms the less advances approach that is based on counts of how often sister nodes swap. We therefore employ a more advanced machine learning method for learning the reordering rules.

The last two approaches are restricted in the word alignment information they can utilize. They discard sequences that overlap on the target side in training data (crossing phrases). This restriction per definition rules out a set of reorderings. For example, a rule such as the French-English *'ne VERB pas'* → *'does not VERB'* cannot be learned, if *'ne pas'* links to *'does not'* as a unit.

For the present work, we also choose to focus on these simple reorderings where two sequences exchange positions. We do this partly to simplify the experiment by narrowing down the learning experience, and partly because we expect these to make more general rules. The example described before would for example need to be very lexicalized.

An final factor that some of the approaches describe is the effect of combining the random distortion-based reordering of the original PSMT ap-

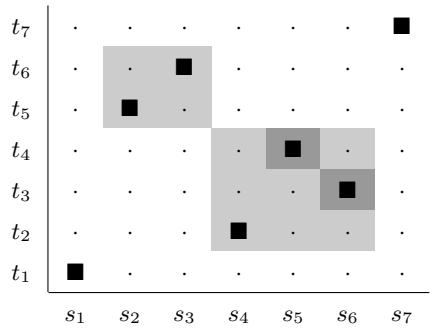


Table 5.2: Abstract example of reordering.

proach with the pre-translation reordering framework. Whether this helps or hurts performance is uncertain. Some experiments report an improvement from this (Habash, 2007b; Li et al., 2007), while others report the opposite (Xia & McCord, 2004; Zhang et al., 2007a). We choose not to include this option to better isolate the effect of the rule-based reordering.

In the next section, we will provide the definition of a reordering that is utilized in the present approach.

5.3 Definition of reordering

In the present work, reordering is defined as two word sequences exchanging positions. These two sequences are restricted by the following conditions:

- **Parallel consecutive:** A sequence must be consecutive and align to a consecutive sequence. Neither of these sequences may link outside the other.
- **Maximal:** They have to be the longest possible consecutive sequences changing place.
- **Adjacent:** They have to appear next to each other on both source and target side.

The sequences are not restricted in length, making both short and long distance reordering possible. Furthermore, they need not be phrases in the

sense that they appear as an entry in the phrase table.

Table 5.2 illustrates reordering in a word alignment matrix. The table contains reorderings between the light grey sequences (s_2^3 and s_4^6)[1] and the dark grey sequences (s_5^5 and s_6^6). On the other hand, the sequences s_3^3 and s_4^5 are e.g. not considered reordered, since neither are maximal, and s_4^5 is not consecutive on the target side.

5.4 Learning rules

In section 3.1, we pointed out that subordination is very important for word order differences between English and Danish. In addition, the sentence position of constituents plays a role. All this information is present in a syntactic sentence parse. A subordinate clause is defined as inside an SBAR constituent; otherwise it is a main clause. The constituent position can be extracted from the sentence start tag and the following syntactic phrases. POS and word form are also included to allow for more specific/lexicalized rules.

Besides including this information for the candidate reordering sequences (left sequence (LS) and right sequence (RS)), we also include it for the set of possible left (LC) and right (RC) contexts of these. The span of the contexts varies from a single word to all the way to the sentence border. Table 5.3 contains an example of the information available to the learning algorithm. In the example, LS and RS should change place, since the first position is occupied by something other than the subject in a main clause.

The information included is much richer than that of Li et al. (2007). They use very sparse single word contexts, and the reordering sequences are only described by the initial, final and head words.

In order to minimize the training data, word and POS sequences are limited to 4 words, and phrase structure (PS) sequences are limited to 3 constituents. In addition, an entry is only used if at least one of these three levels is not too long for both LS and RS, and too long contexts are not included in the set. This does not constrain the possible length of a reordering, since a PS sequence of length 1 can cover an entire sentence.

For the task of extracting rules from the annotated data, we have chosen

[1]Notation: s_x^y means the consecutive source sequence covering word positions x to y.

Level	LC	LS	RS	RC
WORD	<s> today , || today , || ,	he	was driving	home || home . || home . < /s>
POS	<S> NN , || NN , || ,	PRP	AUX VBG	NN || NN . || NN . < /S>
PS	<S> NP , || NP , || ,	NP	AUX VBG	ADVP || ADVP . || ADVP . < /S>
SUBORD	main	main	main	main

Table 5.3: Example of experience for learning. Possible contexts separated by ||.

a rule-based classifier, Ripper (Cohen, 1995; Cohen, 1996) (see section 4.4). The motivation for using Ripper is that it allows features to be sets of strings, which fits well with our representation of the context, and it produces easily readable rules that allow better understanding of the decisions being made. In addition, since Ripper builds on the idea of decision trees, it has a preference for shorter, more covering hypothesis rules (Mitchell, 1997, ch.3). We therefore expect this learning algorithm will lead to highly generalized rules.

The probabilities of the rules are estimated using Maximum Likelihood Estimation based on the information supplied by Ripper on the performance of the individual rules on the training data. The probabilities are smoothed by incrementing the total number of occurrences, as a means for biasing towards more frequent rules. For example, Ripper might inform that a given rule is successful 423 times of the 504 times it applied to the learning data. This provides a probability of 83.8% (423/(504+1)) for that rule. When converted to logarithmic values, these probabilities are easily integratable in the log-linear PSMT model as an additional parameter by simple addition.

5.5 Rule analysis

5.5.1 English-Danish rules

We use different data sets to examine the effect of three relevant parameters:

1. the method employed to obtain the word alignment (*manual* vs. *GDF*),

2. the domain of the training data (*CDEDT* vs. *EP*), and

3. the size of the training data (*100K*, *300K*, and *772K*).

For hand-aligned data (*manual*), we used the Copenhagen Danish-English Dependency Treebank (*CDEDT*) (Buch-Kromann et al., 2007). The annotation guidelines for the manual word alignment process is described in (Elming, 2005). 5478 sentences from the news paper domain containing 112K English words and 100K Danish words. We also create an automatic

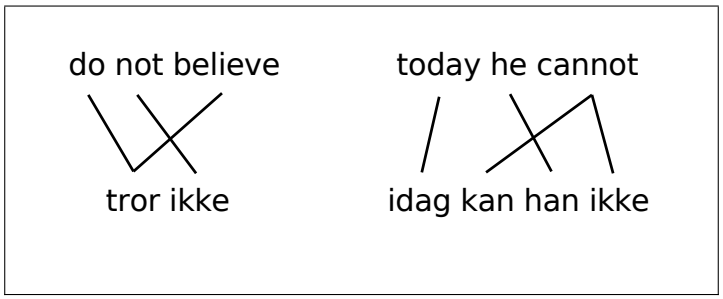


Figure 5.1: Examples of crossing phrases that are excluded as learning data. The left one is a problem, the right one is not.

alignment for these using the GDF algorithm described in section 2.2.2 to symmetrize bi-directional GIZA++ alignments (Och & Ney, 2003) (*GDF*).

The CDEDT data are news paper articles and therefore out of domain for the SMT system used in the next chapter. This system is trained on European Parliament proceedings (*EP*). We therefore also use the English-Danish Europarl corpus version 3 (Koehn et al., 2005). Together, the CD-EDT and EP data provide the out-of-domain and in-domain settings.

In order to test the effect of training data size, we include data sets of three sizes from EP; roughly the same size a CDEDT (*100K* words), three times this size (*300K* words), and the maximal size we are able to train on with Ripper (*772K* words). These data sets are excerpts from a GDF alignment of the entire EP corpus ($\sim$30M words). The GDF alignment of CDEDT is also aligned together with these data. The English side of all data sets is parsed using a state-of-the-art statistical English parser (Charniak, 2000). All rules learned are listed in appendix A.

Table 5.7 shows the statistics for the feature set used to learn English-Danish reordering rules. As can be seen, the set is very skewed distributed, since only a very small percentage of the conditions met in the training data warrant a reordering (0.17%-0.29%).

The table also contains statistics on effect of the restriction to omit reorderings with crossing phrases. Since we in this experiment make no restrictions on which source sequences can change place, the only reorderings that are excluded, are the ones where a single word links to an inconsecutive sequence.

Training data	Feature vectors in total	Reordering feature vectors	Excluded reorderings	Percentage of feature vectors are reorderings	Percentage of reorderings are excluded
CDEDT 100K manual	1,382,524	3,699	1,508	0.27%	29.0%
CDEDT 100K GDF	965,465	1,684	5,104	0.17%	75.2%
EP 100K GDF	832,113	2,427	4,892	0.29%	66.8%
EP 300K GDF	2,367,821	6,900	14,842	0.29%	68.3%
EP 772K GDF	5,500,000	15,906	49,815	0.29%	75.8%

Table 5.4: Statistics of the feature set used for rule learning. *Feature vectors in total* show the total number of vectors trained on. *Reordering feature vector* shows how many of these represent reorderings. *Excluded reorderings* show how many reordering features are not included due to the crossing phrases restriction.

Figure 5.1 illustrates two such cases; one where a target word links to an inconsecutive source sequences (*'do believe → tror'*), and one in the other directions (*'cannot → ikke kan*).

The exclusion of the phenomenon on the right with the inconsecutive target sequence is not a problem in a pre-translation reordering framework. There is simply no way of reordering the source sentence, so it gets the word order of the target sentence. *he* would have to move inside *cannot*, and there is no obvious gain from moving it to the other side of *cannot*.

We therefore only consider the exclusion of the left one with the inconsecutive source sequences a lack in our approach, since moving *believe* to the left of *not* would create a word order that is consistent with Danish.

In the hand-aligned data, more than 1/4 of the reorderings cannot be learned due to this restriction. This is clearly problematic for the present approach compared to fully covering approaches.

It is also striking how much the alignment method shines through here. The ratio between included and excluded reorderings flips, when using GDF instead of manual alignments. Now almost 3/4 of the reorderings are omitted. This reflects the lower quality of the GDF alignments, which contain a lot more crossing links.

If we compare to the manual alignment, most of the excluded reorderings must contain crossing links that are wrong, otherwise they would also have been excluded with manual alignment. Therefore, the restriction may actually help when working with GDF aligned data, since it filters out a lot of reordering noise due to bad alignment. This is backed by manual inspection of the omitted reorderings. It may be for the same reason that PSMT is very robust to bad word alignment, since the worst links are simply excluded by this restriction. We leave this question open for future work and turn to the rules that were learned.

Based on the feature set described in table 5.7, rules were extracted using Ripper. Perhaps because of the very skewed distribution of the feature set, Ripper was unable to learn from a large part of the reorderings in the learning data. Of the reorderings located in the CDEDT data, Ripper was only able to learn 47% with either alignment, and on the EP data, Ripper only learned 15-16% of the reorderings in the feature set. The rest of the reorderings were classified as not being reorderings.

This further narrows down the coverage of the rules. Especially, the manually aligned CDEDT data is very reliable. For these, we are certain that the rule set is only able to handle less than half of the reorderings that actually occur. We now take a closer look at the rules that were learned.

Table 5.5 reveals statistics on the rule sets that were learned for English-Danish reordering. It is very satisfying, that all data sets have learned the three major reorderings described in section 3.1.1; the verb second phenomenon and placement of adverbials in main and subordinate clause. As can be seen, most of the rules learned concerned different ways of identifying contexts where a reordering should occur due to the verb second nature of Danish. The additional rule on currency describes a difference in notational practice, where currency is written after the amount in Danish, while it is the other way around in English. Since the training data however only includes Danish Crowns, the rule was lexicalized to 'DKK'.

To illustrate the rules, table 5.6 shows a few rules learned from the hand-aligned data (see appendix A for a complete list). The first three rules deal with the verb second phenomenon[2]. The only difference among these is the left context. Either it is an initial prepositional phrase, a subordinate clause or an adverbial. These are three ways that the algorithm has learned to identify the verb second phenomenon conditions. Rule 4 is interesting in that it is lexicalized. In the learning data, the Danish correspondent to 'however' is most often not topicalized, and the subject is therefore not forced from the initial position. As a consequence, the rule states that it should only apply, if 'however' is not included in the left context of the reordering.

Rule 21 handles the placement of adverbials in a subordinate clause. Since the right context is subordinate and a verb phrase, the current sequences must also be subordinate. In contrast, the last rule deals with adverbials in a main clause, since the left context noun phrase is in a main clause.

A problem with the hand-aligned data used for learning rules is that it is out of domain compared to the Europarl data used to train the SMT system. The hand-aligned data is news paper texts, and Europarl is transcribed spoken language from the European Parliament. Due to its spoken

[2] We merged all finite verb POS tags to one tag FVF.

Rule set	Total number of rules	Verb second	Main clause sentence adverbials	Subordinate clause sentence adverbials	Currency placement
CDEDT 100K manual	27	22	4	1	1
CDEDT 100K GDF	19	13	3	2	1
EP 100K GDF	14	12	1	1	0
EP 300K GDF	20	18	1	1	0
EP 772K GDF	15	13	1	1	0

Table 5.5: Statistics of the rule sets learned. CDEDT 100K manual specified rules does not sum to the total number of rules, since one rule covers both verb second and sentence adverbials in main clause.

No	LC	LS	RS	RC	Prob.
2	PS: <s> PP ,	PS: NP	POS: FVF		83%
3	PS: SBAR ,	PS: NP	POS: FVF		82%
4	PS: ADVP , ! WORD: however ,	PS: NP	POS: FVF		70%
21		POS: FVF	POS: RB	PS: VP SUB: SUB	71%
8	PS: <s> NP SUB: MAIN	PS: ADVP	POS: FVF		62%

Table 5.6: Example rules and their application statistics on the test set, when using the reordering approach described in chapter 6. Redundant information has been removed.

nature, Europarl contains frequent sentence-initial forms of address. That is, left adjacent elements that are not integrated parts of the sentence as illustrated by example (11).

This is not straightforward, because on the surface these look a lot like topicalized constructions, as in example (12). In topicalized constructions, it is an integrated part of the sentence that is moved to the front in order to affect the flow of discourse information. This difference is crucial for the reordering rules, since 'i' and 'have' should reorder in (12), but not in (11), in order to get Danish word order.

(11) mr president , i have three points .

(12) as president , i have three points .

When translating the development set, it became clear that many constructions like (11) were reordered by a rule. Since these constructions were not present in the hand-aligned data, the learning algorithm did not have the data to learn this difference.

We therefore included a manual, lexicalized rule stating that if the left context contained one of a set of titles (*mr*, *mrs*, *ms*, *madam*, *gentlemen*), the reordering should not take place. To a great extent, the rule eliminates the problem. Since the learning includes word form information, this is a rule

that the learning algorithm in theory should be able to learn. However, the word context in the present experiment is made up of sequences of words bordering on the reordering sequences. In example (11) and (12), this means that the relevant sequences would be *'mr president ,'* and *'as president ,'*. The learner would not have information that *mr* appeared in the context, only the entire string. Therefore, the it is not learned from the in-domain EP data either. However, a few of these rules contain the information that if *'president ,'* is in the context, then the words should not reorder.

The above examples also illustrate that local reordering (in this case as local as two neighboring words) can be a problem for PSMT, since even though the reordering is local, the information about whether to reorder or not is not necessarily local.

5.5.2 English-Arabic rules

The purpose of this experiment is two-fold; first, we seek to investigate the portability of this approach to less similar languages, and second, we wish to examine how well the *combination* alignment method described in chapter 4 is suited as basis for learning reordering rules. Mainly because the AER metric evaluates it to be much closer to the hand-alignment, than the GDF alignment is. All rules learned are listed in appendix A.

Rules are extracted from the IBMAC corpus described in section 4.5.1. This data set comes in a hand-aligned version, and from the experiments in chapter 4, we have a combination alignment and a GDF alignment of 6.6K of the sentences from sections 4.5.5. We use these 6.6K sentences (179K/146K words) for rule extraction. The English side is parsed using a state-of-the-art statistical English parser (Charniak, 2000).

Since we want to use the hand-aligned data, we are again forced to use the simple AR tokenization scheme for Arabic. This may prove detrimental to the reordering experiment, since some of the reordered elements are clitically attached as described in section 3.1.2, and these are not segmented out in the AR scheme. Also it is generally accepted that higher Arabic segmentation leads to better translation (Lee, 2004; Habash & Sadat, 2006).

Table 5.7 shows the statistics for the feature set used to learn English-Arabic reordering rules. The distribution is not as skewed as for the

Training data	Feature vectors in total	Reordering feature vectors	Excluded reorderings	Percentage of feature vectors are reorderings	Percentage of reorderings are excluded
Manual	1,444,560	18,578	9,049	1.3%	33.8%
Combination	1,414,983	18,617	9,343	1.3%	33.4%
GDF	1,332,925	26,407	6,184	2.0%	19.0%

Table 5.7: Statistics of the feature set used for rule learning. *Feature vectors in total* show the total number of vectors trained on. *Reordering feature vector* shows how many of these represent reorderings. *Excluded reorderings* show how many reordering features are not included due to the crossing phrases restriction.

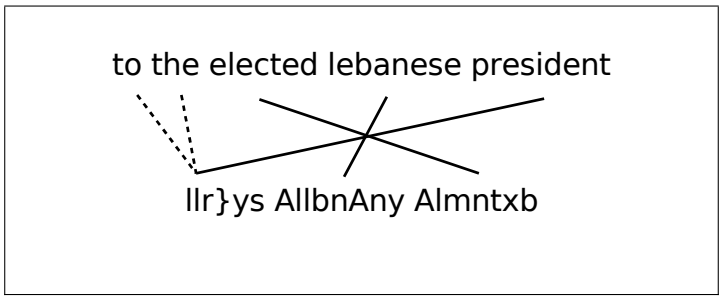


Figure 5.2: Example of a complexly aligned English-Arabic segment. The dotted line are links that are present in the manual alignment, but not included in the GDF alignment.

English-Danish experiment, since this task involves more reordering. Here, they make up around 10 times as much of the feature set (1.3%-2.0%). Still, this must be regarded as a skewed distribution. An interesting point of the table is that the statistics for the combination alignment look very similar to the manual alignment.

Again, we report numbers for the effect of excluding data based on the phrase consecutiveness constraint. For the manual alignments, the excluded reorderings are at the same level as for Danish (29.0% for Danish, 33.8% for Arabic).

For the GDF alignments on the other hand, the statistics behave very differently from Danish. For Danish, more than twice as big a percentage of reorderings were excluded with GDF alignments than with manual alignments, but for Arabic, it is the other way around. Here, the manual alignment leads to almost twice as big a percentage of the reorderings being excluded as is the case for GDF alignments.

A reason for this may be the low level of Arabic segmentation used in these experiments. This means that the English sentences in general contain much more words than the corresponding Arabic. The English sentences contains 23% more words, which is almost 5 words per sentence. Compared to Danish, where the English sentences contain 12% more words, which is about 2 words per sentence. However, with fewer target words, each one will on average link to more source words, and this would expectedly lead to more inconsistent source sequences, since these only appear when a single target word aligns to more than one source word. The reason for this may be that the high complexity of an alignment between

low-level segmented Arabic text and English is not produced by GDF that tries to create the least distorted and complex alignment. An example of this is shown in figure 5.2. The dotted line are links that are present in the manual alignment, but not included in the GDF alignment. Here, the two English words *to* and *the* do not aligned to anything.

When learning the rule sets with Ripper, we saw the same pattern as for English-Danish. Ripper was unable to learn from a large part of the reorderings that were available. Here, we also expect the skewed distribution of the feature set to play its part. Again, Ripper learned less than half the reorderings (manual: 44%, combination:18%, GDF:49%). Especially disturbing is the very low percentage with the combination alignment. We now take a closer look at the rules that were learned.

Table 5.8 shows the statistics of the rule sets that were learned for English-Arabic reordering. Of the reorderings described in section 3.1.2, both the verb-before-subject and the adjective-after-noun reorderings were learned, but as expected the possessive-pronoun-after-noun reordering could not be learned within this preprocessing scheme. Additionally, a lot of rules on moving noun-after-noun and a rule placing the final quote before the period were learned.

A few of the rules that were learned from the manual alignment, are show in table 5.9 (see appendix A for a complete list). The first two rules handle the placement of the finite verb in Arabic. Rule 16 states that if a finite verb appears in front of a subordinate clause, then it should be moved to sentence initial position with a probability of 68%. Due to the restrictions of sequence lengths, it can only swap across maximally 4 words or a sequence of words that is describable by maximally 3 syntactic phrases. The SBAR condition may help restrict the reordering to finite verbs of the main clause. This rule and its probability goes well with the description given in sections 3.1.2, since VSO order is not obligatory. The subject may either not be expressed, or it may appear in front of the verb. This is even more obvious in rule 27, which has a probability of only 43%.

Rules 11 and 1 deal with the inverse ordering of adjectives and nouns. The first is general but uncertain, the second is lexicalized but certain. The reason for the low probability of rule 11 is primarily that many proper names have been mis-tagged by the parser as either JJ or NN, and to a

Rule set	Total number of rules	Adjectives to the right of noun	Noun to the right of noun	Adjective to the right of adjective	Verb initial	Quotes before period
Manual	61	49	10	19	3	1
Combination	20	19	7	0	0	0
GDF	39	34	17	7	1	1

Table 5.8: Statistics of the rule sets learned. Note that one rule may cover more categories.

No	LC	LS	RS	RC	Prob.
16	WORD: <s>		POS: FVF	PS: SBAR	68%
27	WORD: <s>	PS: NP	POS: FVF		43%
11	POS: IN	POS: JJ	POS: NN		46%
1	! POS: JJ	POS: JJ	WORD: president		90%
37	! POS: NN ! POS: JJ	POS: NN	POS: NNS	POS: IN	71%

Table 5.9: Example rules and their application statistics on the test set, when using the reordering approach described in chapter 6. Redundant information has been removed.

lesser extend that the rule should often not apply if the right context is also an NN. Adding the latter restriction narrows the scope of the rule but would have increased the probability to 54%.

Rule 1, on the other hand, has a high probability of 90%. It is only restricted by the condition that the left context should not be an adjective. In these cases, the adjectives should often be moved together, as is the case with *'the south african president → Alr}ys Aljnwb Afryqy'* where *'south african'* is moved to the right of *'president'*.

Finally, rule 37 presents a phenomenon that was not described in section 3.1.2. Here a singular noun is moved to the right of a plural noun, if the right context is a preposition, and the left context is neither an adjective nor a singular noun. This rule handles compound nouns, where the modifying function of the first noun often is hard to distinguish from that of an adjective. The left context restrictions server the same purpose as the left context in rule 1; these should often be moved together with the singular noun. The function of the right context is harder to explain, but without this restriction, the rule would have been much less successful; dropping from a probability of 71% to 51%.

5.6 Conclusion

In this chapter, we have introduced a novel approach to automatically learning reordering rules. One ambition in connection with this was to

only learn a very general subset of rules. Compared to the amount of rules produced by previous approaches, this was accomplished.

The approach was not as general as hoped for. For example, in the English-Danish experiment based on hand-aligned data, 22 of 27 rules described the same reordering under different circumstances. Nonetheless, this is much less than what can be expected from the other approaches.

In addition, we created an approach that integrates a much richer set of features, and that is able to select among these to pick out the ones that are important for a given phenomenon.

A problem with the learning algorithm utilized was that it was able to learn less than half of the reorderings available in the training data. We believe to a large part that it had problems handling the very skewed distribution of the feature set. In the future, we are interested in exploring other learning algorithms that may provide wider coverage. A Maximum Entropy approach in line with (Li et al., 2007) may be such an option.

As a compromise in the pursuit for general rules, we diverged from a fully covering approach by excluding reorderings containing overlapping target sequences. However, we are interested in expanding to a fully covering approach in the future, since we believe this will further strengthen the approach. In the hand-aligned data, around 1/3 of the reorderings were omitted due to this restriction.

An interesting aspect in this context is that the experiments indicated that when using automatically aligned data, this constraint may help clean up the data by ruling out a large amount of reordering noise. The constraint may therefore strengthen rule learning under these circumstances. We leave this question open for future investigation.

Finally, investigations done by Fox (2002) indicate that a flatter tree structure provides a better basis for rule extraction, and a dependency structure may be better than a phrase structure. This would be interesting to examine in future experiments.

In the next chapter, we will turn to the exploitation of these rules in a novel approach to reordering in PSMT.

Chapter 6

Integrating syntactic reordering in phrase-based SMT

In chapter 3, we described previous pre-translation reordering solutions to the reordering problem in PSMT. We also noted a set of problems with these. Most notably, the problems in assigning a score to a reordered source sentence based on experience from the relation between source and target word order. We will briefly sum up this problem.

In all previous pre-translation reordering approaches, the relation between source and target word order is used to model the relation between source and source with target (ST) word order. The basic assumption in these approaches has been that providing the source sentence with target word order will reflect on the target sentence in translation. This assumption, however, does not always hold. Phrase internal reorderings can lead to reorderings appearing between the ST and target sentence even if additional external reordering is excluded. This means that a hypothesis can be assigned a score that is in conflict with the learning experience.

In this chapter, we introduce a novel approach to syntax-based reordering in PSMT that overcomes these problems. We examine the approach in the context of a pre-translation reordering framework, but the approach could also be integrated into other frameworks. The approach is probabilistic and non-deterministic, and hypothesis word orderings are evaluated based on the order of the target words. This not only leads to a theoretically more satisfying model of reordering that provides superior integra-

tion in PSMT, but we also prove that it brings improved translation quality to PSMT. We call the approach *source position target order* (SPTO) scoring.

We first describe the baseline PSMT system used in the experiments. Then, we go into how this was extended for pre-translation reordering. In section 6.3, we describe the novel SPTO scoring approach. This is followed by first English-Danish and then English-Arabic experiments evaluating, examining, and analyzing the approach. In the final section, we discuss and conclude on the experiments.

6.1 The PSMT system

The baseline is the PSMT system used for the 2006 NAACL SMT workshop (Koehn & Monz, 2006) with phrase length 3 and a modified Kneser-Ney smoothed (Chen & Goodman, 1998) trigram language model trained using the SRILM toolkit (Stolcke, 2002). The alignment symmetrization heuristic used is *grow-diag-final* as described in section 4.3.

The decoder used for the baseline system is Pharaoh (Koehn, 2004a), which uses the following information sources as described in section 2.2.2: bidirectional phrase translation models, bidirectional lexical weighting models, phrase and word penalty, a target language model, and a distance-penalizing distortion model.

We consider Pharaoh with default distortion limit our baseline system, since it allows for reorderings of unlimited distance same as our reordering approach. This is, however, not very important, since we also report on Pharaoh with distortion limit 4 and monotone decoding.

Due to the random reorderings required by the simple distortion model, Pharaoh must evaluate all possible source word orders in translation, which is an NP-complete problem (Knight, 1999). It therefore employs a beam search algorithm (Jelinek, 1998) to reduce the size of the search space in the following ways;

First of all, for any two hypotheses that look the same to future decoding actions, only the most probable is kept. This is the case if the last (n-1) target words are the same (which is all a language model of order n sees), the position of the last covered source word is the same (which is all the distortion model sees), and they cover the same source words (which is all

the rest of the parameters see). This pruning is without loss.

Secondly, hypotheses are stored in stacks based on the number of words they cover. The size of these stacks is restricted to a maximal amount of hypotheses, and if a hypothesis is much less likely than the best hypothesis, it is not included. The retained hypotheses are selected based on their probability so far and an estimated future probability. If the latter is not included, the algorithm has a tendency to place the difficult/unknown parts at the end of the sentence. Since it is only an estimation, the search algorithm is no longer guaranteed to find the best path.

As proposed by (Och, 2003), the parameters of the PSMT system should be optimized towards producing the best translation. We optimize towards the BLEU metric using the Downhill Simplex algorithm (Nelder & Mead, 1965). This algorithm is suited for large dimensional optimization tasks such as the PSMT systems described here, which have 8 weights to optimize. It locates a local minimum.

The algorithm creates a simplex, which is a polytope with N+1 vertices (e.g. a triangle on a two-dimensional plane). Each vertex is then evaluated based on its coordinates in the N-dimensional space. Here, the evaluation is provided by using the coordinates as parameter weights in the PSMT system when translating a small tune set of 500 sentences. This translation is assigned a BLEU score, and one minus this score is returned, since we are looking for a minimum. Using a set of rules, the highest valued vertex is modified, and another translation is produced. This continues until the algorithm converges.

6.2 Pre-translation reordering

The present experiments are set in the context of a pre-translation reordering framework. The word order scoring is, however, done on the target sentence as opposed to previous approaches. The rule application is therefore carried out in two separate stages:

1. Pre-translation reordering based on the rules.

2. SPTO scoring of the target word order according to the rules.

In this section, we describe stage 1, while the novel SPTO scoring is described in the next section.

The first stage — pre-translation reordering — is done in a non-deterministic fashion by generating a word lattice as input in the spirit of e.g. (Zens et al., 2002; Crego & Mariño, 2007; Zhang et al., 2007b). This way, the system has both the original word order, and the reorderings predicted by the rule set. The different paths of the word lattice are merely given as equal suggestions to the decoder. They are in no way individually weighted.

Figure 6.1 shows a source sentence, a rule, and the resulting word lattice created for translation. The example is taken from the English-Danish test set, and the rule is learned from the English-Danish CDEDT hand-aligned corpus. The word lattice has been abbreviated and broken up in two, due to space restrictions, and the reordered verbs are highlighted.

The rule concerns the verb second nature of Danish. It states that an NP followed by a finite verb should change places, if the NP is preceded by an adverbial. This rule applies twice to the source sentence based on the provided parse, since the initial adverbial *'obviously'* precedes both the NP *'the work'*, which is followed by the finite verb *'carried'*, and the NP *'the work carried out by each of the rapporteurs and by the parliamentary committees'*, which is followed by the finite verb *'will'*. This leads to three paths in the word lattice, where the bottom path moving *'will'* in front of the long NP provides the best Danish word order. The other reordering that moves *'carried'* to the left of the short NP, is wrong. This owes to a bad parse, since *'carried'* is not a finite verb in this sentence but a passive participle in a relative clause. The rule should therefore not have applied. The final path is the original source word order.

Since Pharaoh does not accommodate these needs, we use our own decoder, which — except for the reordering model — uses the same knowledge sources as Pharaoh. Its translations are comparable to Pharaoh when doing monotone decoding. The search algorithm of our decoder is similar to the RG graph decoder of (Zens et al., 2002).

Since we do not allow random external reordering in our experiments, this is restricted to the paths of the word lattice. It is therefore possible to limit the search to a monotone setting, i.e. one where the words are simply

SOURCE

obviously$_{RB}$ [$_{NP_1}$[$_{NP_2}$ the work]$_{NP_2}$ carried$_{FVF}$ out by each of the rapporteurs and by the parliamentary committees]$_{NP_1}$ will$_{FVF}$ serve a useful purpose in providing fuel for our own discussions .

RULE

LC	LS	RS	RC	Prob.
POS: RB	PS: NP	POS: FVF		74.0%

WORD LATTICE

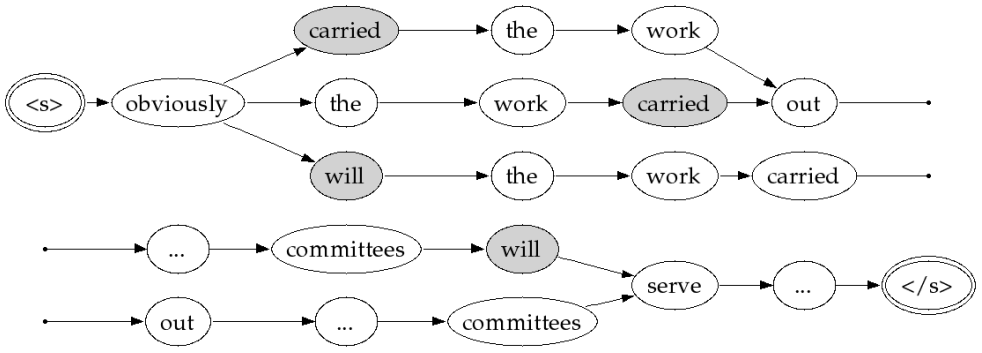


Figure 6.1: Example source sentence reordered to a word lattice. In the source sentence, relevant syntactic phrases are indicated with subscripted square brackets, and POS is subscripted the word. The word lattice is abbreviated and split in two due to space restriction, and the nodes of the reordered verb are highlighted.

translated in the order they appear in the lattice. In other words, decoding follows the lattice one node at a time, building hypotheses covering up to that node. This means that the search space is reduced substantially and less pruning is needed.

We only employ a pruning mechanism similar to the first one used in Pharaoh, where for any two hypotheses that look the same to future decoding actions, only the most probable is kept. This condition is here met, when the target string of two hypotheses at the same node in the lattice share the same last (n-1) words, since these also cover the same source words. None of the less probable hypotheses can possibly outperform this one later on. This is because the maximal context evaluating an extension of this hypothesis, is the history (n-1-gram) of the first word of the extending phrase. This pruning is without loss, and the search algorithm is therefore optimal.

The nodes are processed in the decoder one at a time in an order defined by the number of words covered so far. That is if a node has a path leading to it that passes 4 words, and another node has one passing 7, then the first node is processed earlier than the other. This means that when the decoder processes a node, we are certain that all nodes leading to this node have already been processed. This is not only important to the monotone decoding, but also to the assumption that supports the pruning mechanism.

6.3 SPTO scoring

The second stage in rule application in our experiments is the SPTO scoring. The basic functionality of this approach is to motivate certain word orders in the translation based on the probabilities supplied by the rules. By evaluating the word order of the resulting translation instead of an intermediate product, it overcomes the problems of previous approaches.

A hypothesis is assigned a score based on how its word order correlates with the predictions made by the rules given the source sentence. The evaluation is made possible by keeping track of which source words the hypothesized target words originate from. This information is obtained via phrase internal word alignments. We call this *source position target order* (SPTO).

To keep track of the word order, each decoder hypothesis contains two parallel strings; a target word string and its SPTO string. In order to access this information, each phrase table entry is annotated with its internal word alignment. This is available as an intermediate product from phrase table creation. If a phrase pair has multiple word alignments, the most frequent is chosen.

We will use the problematic example from figure 3.1 on page 57 to exemplify the SPTO scoring. This example illustrated the problems with a source order (SO) scoring approach such as the previous approaches described in section 3.2.2 on page 53. The example is repeated in full in figure 6.2 together with relevant information.

First, the source sentence is shown with linguistic annotation where relevant. Then we show the rule that applied to the example. The rule is from the set learned on the hand-aligned CDEDT data. It states that an adverbial phrase should change place with the following finite verb, if the left context is a sentence initial noun phrase. In addition, this should only take place if the left context is part of a main clause, which in practice means that this rule applies to adverbials in the main clause (see section 3.1.1).

The following two lines show what effect the reordering has on the source sentence; both in word forms and in their source positions. This position information is used differently in decoding by the SO scoring and the SPTO scoring. If the SO scoring translates a source sentence with the position order *'3 2'*, then the rule is thought to be satisfied, and the hypothesis is assigned the probability 62.8%. In figure 6.2, any translation stemming from path 2 in the word lattice obeys this requirement, and as can be seen from the decoding table, these are therefore scored as satisfying the rule. This is incorrect, since *'derfor' (therefore)* ends up at three different places in the output based on path 2. On the other hand, every translation stemming from path 1 is scored as not satisfying the rule, even though H1 in fact has the desired word order.

SPTO scoring avoids this problem. Here, the hypothesis SPTO must contain the position order *'3 2'* for the rule to be considered satisfied. As can be seen from the decoding table, it is these hypotheses where *'derfor' (therefore)* has the desired position in relation to *'mener' (think)*. It is irrelevant which path was chosen.

SOURCE

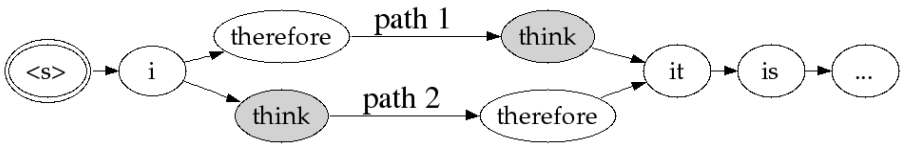
$[_{NP}$ i$]$ $[_{ADVP}$ therefore$]$ think$_{FVF}$ it is extremely important for the european union to get fully involved with this country , and with the situation in peru , for example by maintaining an orderly dialogue with the forces of opposition .

RULE

LC	LS	RS	RC	Prob.
PS: <s> NP SUBORD: main	PS: ADVP	POS: FVF		62.8%

Rule predicted reordering: *therefore think* → *think therefore*
Expressed in source positions: 2 3 → 3 2

WORD LATTICE



DECODING

Hyp	Path	Target words	SPTO	SO score	SPTO score
H1	1	*jeg mener derfor*	1 3 2	-	+
H2	1	*jeg derfor mener*	1 2 3	-	-
H3	2	*jeg mener derfor*	1 3 2	+	+
H4	2	*jeg derfor mener*	1 2 3	+	-
H5	2	*jeg mener derfor er det*	1 3 2 5 4	+	+
H6	2	*jeg mener det er derfor*	1 3 4 5 2	+	-

Figure 6.2: Illustration of the decoding process with SPTO integrated. In the source sentence, relevant syntactic phrases are indicated with subscripted square brackets, and POS is subscripted the word. In the word lattice, nodes of reordered verbs are highlighted, and the paths are called 1 and 2 for reference. Finally, possible hypotheses in decoding are shown with relevant information. The last two columns indicate whether the relevant scoring approach sees a reordering (+) or not (-).

Source phrase	Target phrase	SPTO	Weighted prob
think therefore	*derfor mener*	2 1	4.3%
	mener derfor	1 2	5.3%
therefore think	*derfor mener*	1 2	8.5%
	mener derfor	2 1	10.8%
therefore it is	*det er derfor*	2 3 1	6.2%
	derfor er det	1 3 2	13.1%

Table 6.1: Excerpt of the phrase table used in the English-Danish experiment containing phrases relevant to the example in figure 6.2. The probability of the phrase table entry is weighted by the parameter weights used for the SO system in the experiment.

As table 6.1 shows, the phrase table does not provide much help on the ordering of *'derfor' (therefore)* and *'mener' (think)*. The first 4 entries show that both orderings are common in Danish, since the probability of translating into any of the word orders is very similar given the source word order. In addition, path 2 provides the possibility of moving *'derfor'(therefore)* further to the right via phrase internal reorderings. This is exemplified by the second to last entry.

That the SO scoring approach in our experiments has chosen H6, which uses the second to last phrase pair in table 6.1, shows the enormous influence of the language model. Even though H5 is more than twice as probable according to the translation model, the language model forces H6 through. It also perfectly illustrates the problem with the SO scoring approach. H6 has been rewarded for reordering *'therefore'* and *'think'* as proposed by the rule, but due to a phrase internal reordering that is invisible to the SO scoring, *'therefore'* ends up changing place with *'think it is'*. In other words, the hypothesis is rewarded for following a rule that it in fact does not follow.

Since more than one reordering is often predicted in the same area, we group the reorderings based on the axis they reorder around. This is exemplified by figure 6.3. In this example, there are two reordering groups around axis 1 and 2. We will focus on the second, and therefore only provide probabilities for axis 2 rules. If any of the axis 2 rules is satisfied by a hypothesis, the hypothesis is assigned the probability of this rule. If on

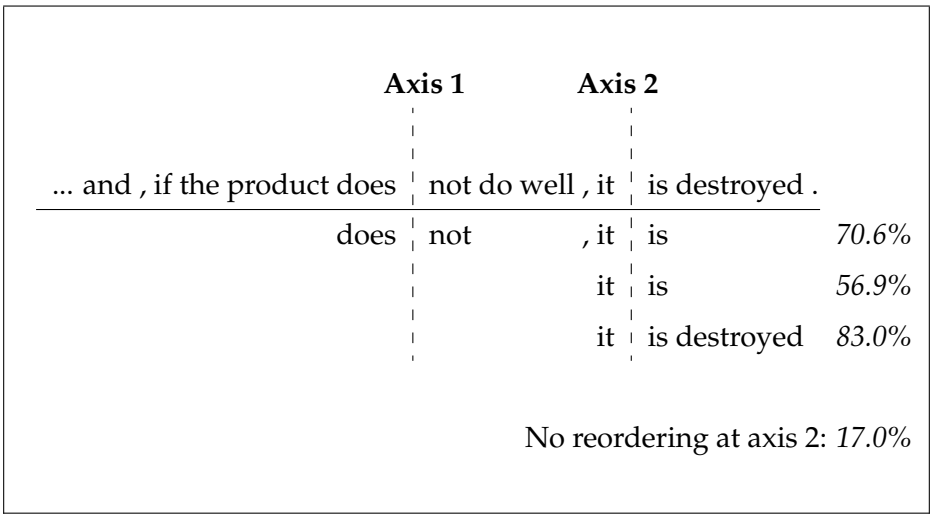


Figure 6.3: Example of group reorderings based on their reordering axes. Source sentence above and reordering suggestions below line. Axes illustrated by dotted line, which separates the left and right reordering sequence. Probabilities are only provided for axis 2 rules.

the other hand no axis 2 rule is satisfied, the hypothesis is assigned the probability for no reordering at axis 2. This probability is defined as the probability of not satisfying the most probable rule. In this example, the most probable rule is 83.0% sure. Therefore, the probability of not doing any reorderings around axis 2 is set to 17.0% (100.0%-83.0%).

Phrase internal reorderings at other points of the sentence, i.e. points that are not covered by a rule, are not judged by the reordering model. Our rule extraction does not learn every possible reordering between the two languages, but only the most general ones. If no rule has an opinion at a certain point in a sentence, the decoder is free to chose the phrase translation it prefers without reordering cost.

Separating the scoring from the source language reordering also has the advantage that the SPTO scoring in essence is compatible with other approaches such as a traditional PSMT system. We will, however, not examine this possibility further in this thesis.

Data	Sentences	English words	Danish words
Training	1.1M	31M	30M
Test	11K	330K	310K
Development	10K	287K	265K
Tuning	500	8.3K	7.6K

Table 6.2: Statistics of data used for the English-Danish experiments.

6.4 Evaluation

6.4.1 English-Danish SPTO experiment

Data

The system was trained on the English and Danish part of the Europarl corpus version 3 (Koehn et al., 2005). Fourth quarter of 2000 was removed in order to use *the common test set*[1] of 11K sentences with one reference (330K English and 310K Danish words) for testing. In addition, fourth quarter of 2001 was removed for development purposes. Of these, 10K (287K English and 265K Danish words) were used for various analysis purposes, thereby keeping the test data perfectly unseen. 500 sentences (8.3K English and 7.6K Danish words) were taken from the development set for tuning the decoder parameter weights. In total, 1.1M sentences containing 31M English words and 30M Danish words were left for training the phrase table and language model.

Results and discussion

The SPTO reordering approach is evaluated on the 11K sentences of *the common test set*. Results are listed in table 6.13 along with results on the development set. We also report on the *swap subset*. These are the 4690 sentences in the test set where a reordering was proposed by a rule, resulting in an internal or external reordering or a rejection of the reordering. The remaining 6679 sentences were not influenced by the SPTO reordering approach.

[1] Following the definition at http://www.statmt.org/europarl/

	NIST			BLEU		
System	**Dev**	**Test**	**Swap**	**Dev**	**Test**	**Swap**
Pharaoh free	7.2225	7.1182	6.7209	26.23	25.17	23.52
Pharaoh dl4	7.2317	7.1309	6.7389	26.39	25.35	23.78
Pharaoh monotone	7.2186	7.1328	6.7430	26.44	25.50	23.93
no scoring	7.1844	7.1068	6.7248	26.66	25.62	24.14
SO scoring	7.1965	7.1180	6.7431	26.78	25.77	24.46
SPTO scoring	7.1985	7.1193	6.7453	26.79	25.79	24.49

Table 6.3: Automatic evaluation scores for different systems. Significance is indicated with color codes. The SPTO system is significantly better than the light grey cells, and significantly worse than the dark grey cells. Both at a 95% confidence level.

We report on the baseline PSMT system both 1) without reordering restrictions, 2) a distortion limit of 4 (see section 2.2.2), and 3) no distortion allowed (monotone), 4) a system provided with a rule reordered word lattice but no scoring in the spirit of e.g. (Crego & Mariño, 2007), 5) the same system but with an SO scoring in the spirit of e.g. (Zhang et al., 2007b; Li et al., 2007), and finally 6) the same system but with the SPTO scoring. For the pre-translation reordering systems, we report on systems using the best performing rule set, which for English-Danish is the one trained on manually aligned data.

An interesting aspect of the results is that Pharaoh performs best under monotone conditions. The more reordering allowed, the worse performance. In other words, the length penalizing distortion model only hurts performance for English-Danish translation. This reveals the difficulty of the reordering task for these closely related languages. If reorderings are not very well motivated, chances are that they will be erroneous.

However, all pre-translation reordering systems increase over the simple monotone decoding. The fact that this is also the case for the simple no scoring system indicates that merely providing the decoder with a syntactically motivated search space is a help. The SPTO approach gets gets the highest score with an increase over the baseline PSMT system of 0.6 %BLEU. All SPTO BLEU scores are significantly better than all other sys-

tems, with the exception of SO ($p < 0.05$). See section 2.4 for details on significance testing.

The rules are, however, not very productive on the test set. That is, they do not produce very many reordering suggestions. They only apply to 2/5 of the sentences. The relevant set, i.e. the set where the SPTO approach actually applies, is therefore the swap subset. This way, we concentrate on the syntactically motivated SPTO reordering. On this set, the effect of the SPTO approach becomes more outspoken. We achieve an increase in performance of 1.0 % BLEU. Again it significantly outperforms all systems except for SO ($p < 0.05$).

In order to get a better understanding of the relation between the different pre-translation reordering approaches, we focus on the instances where they produce different translations. The three approaches behave very alike. While the SPTO approach on the test set produces a different translation than either Pharaoh system about 9,000 times ($\sim$80% of the sentences), it only differs from the no scoring approach 1898 times (17%), and from the SO approach 475 times (4%).

A major reason for this closeness is that the same parameter weights were used for the three pre-translation reordering systems. On the development set, this factor diminishes the difference in translation from 29% of the sentences to 11% and 43% to 17% respectively. Initially, all systems were optimized individually, but when translating on the development set, we saw that the weights optimized for the SPTO approach provided a marginally better translation for all pre-translation reordering systems. We therefore used these weight for all three systems to get a fairer picture. This also has the advantage that we neutralize optimization as an intervening factor, thereby achieving a clearer comparison of the approaches. For the baseline system, these weights decreased performance, so it retained its original optimization.

Since the small number of differences between the SPTO and SO approaches will be virtually unnoticeable to the scoring metric in the large test set, we also report on the subset of test sentences, where these two differ in translation. This set is interesting, since it provides a focus on the difference between the SO and the SPTO approaches. We call this data set the *diff set*. In table 6.4, we evaluate on it.

System	NIST	BLEU	Avr. human rating
Baseline	6.0085	23.18	3.0 (2.56)
no scoring	6.0233	23.71	3.0 (2.74)
SO scoring	5.9938	23.53	3.0 (2.62)
SPTO scoring	6.0067	23.82	2.0 (2.08)

Table 6.4: Evaluation on the diff set. Average human ratings are medians with means in parenthesis, lower scores are better, 1 is the best score. The SPTO system is significantly better than the light grey cells.

Decoder choice	NO	SO	SPTO
Phrase internal reordering (I)	886 (15%)	401 (7%)	1538 (27%)
Phrase external reordering (E)	1454 (25%)	3846 (67%)	2854 (50%)
Reject reordering (R)	3379 (59%)	1472 (26%)	1328 (23%)

Table 6.5: The reordering choices made based on the three pre-translation reordering approaches for the 5719 reorderings proposed by the rules for the test data. The percentage shows how much a choice makes up of all choices made by the approach.

The BLEU scores on the entire diff set indicate that SPTO is a superior scoring method. To back this observation, 100 sentences are manually evaluated by two native speakers of Danish. The annotators showed reasonable inter-annotator agreement ($\kappa = 0.523, P(A) = 0.69, P(E) = 0.35$). Table 6.4 shows the average ratings of the systems. Means are reported, since they provide more nuance in this case. This shows the SPTO scoring to be significantly superior to the other methods ($p < 0.001$). See section 2.4 for details on agreement and significance testing.

Analysis of reordering statistics

We now provide statistics for the reordering choices made by the pre-translation reordering approaches on the entire test set. Following this, we go into more detail with these choices using a manual analysis of the 117 reorderings proposed by rules in the 100 sentences that were manually evaluated. We also look at the effect of the individual rules used to produce these 117 reorderings.

Table 6.5 shows the effect of the three scoring approaches for pre-translation reordering in decoding. Most noticeable is that the SO scoring is strongly biased against phrase internal reorderings; SPTO uses nearly four times as many phrase internal reorderings as SO. In addition, SPTO is slightly less likely to reject a rule proposed reordering.

It is also striking that the no scoring approach in fact does not utilize the rules very well, rejecting them more often than not. This results in almost half as many internal and external reorderings as the SPTO approach. There is, however, no noticeable bias against either reordering compared to the SPTO approach.

Table 6.6 lists the reordering choices made by the SPTO system compared to the choices made by the other systems. The possible choices are E (phrase external reordering), I (phrase internal reordering), and R (reordering rejected). In the *reorder choice* column are listed the possible combinations of reordering choices for SPTO paired with the other approaches. For example, E – R means that the SPTO approach chose to use an external reordering to comply with the rule-proposed reordering, while the other approach rejected the same reordering. For the baseline system, the choices are merely observations as to whether it satisfies the rules. It does not have this information in translation. This information was not present in table 6.5, since we were not able to extract it automatically.

All approaches are ranked according to how they handled each proposed reordering. This is done in a similar fashion to the manual evaluation. The judgement is based on the translation produced for the subpart of the sentence that the reordering concerns. If a good reordering is made by two approaches, but one leads to a better translation, this one is ranked higher. If no reordering leads to a better translation than a successful reordering, then the better translation is ranked highest. This could for example be the case if a rule suggests an incorrect reordering, and a system correctly rejects it.

Since this set is defined by the differing translations of SPTO and SO, the distribution of reordering choices differ from that of the entire test set. When comparing the distribution for the diff set in table 6.7 with the distribution of the entire set in table 6.5, the main difference is that the SPTO translations contains a very high percentage of internal reorderings com-

Reorder choice	SO scoring			NO scoring			Baseline			Across systems	
	Better	Same	Worse	Better	Same	Worse	Better	Same	Worse	Impr.	Decr.
E – E		8			3						
E – I	4							5		44%	6%
E – R				6	2	1	6		1		
I – E	21	36	7				1				
I – I		1			35		3	38	6	46%	10%
I – R	20	3	4	43	7	7	38	2	4		
R – E	1	2	2		1						
R – I		1								5%	10%
R – R		7			12		1	10	2		
Total	46	58	13	49	60	8	49	55	13	41%	10%

Table 6.6: Manual analysis of reordering choices made by the SPTO approach conducted on the 117 proposed reorderings in the 100 manually evaluated sentences. The choices are either E (external reordering), I (internal reordering), or R (rejected reordering). First choice (pre-dash) is that of the SPTO system, the second (post-dash) is of the other approach. For the baseline system, the choices are merely observations as to whether it satisfies the rules. It does not have this information in translation. The translation of the reordered area is compared to other approaches as either better, same, or worse. Zero counts are excluded for ease of readability. Percentage of SPTO reordering choices that result in improvement/decrease across the other systems is also reported.

Decoder choice	NO	SO	SPTO
Phrase internal reordering (I)	166 (30%)	42 (8%)	383 (70%)
Phrase external reordering (E)	39 (7%)	379 (69%)	95 (17%)
Reject reordering (R)	343 (63%)	127 (23%)	70 (13%)

Table 6.7: The reordering choices made based on the three pre-translation reordering approaches for the 548 reorderings proposed by the rules for the diff set. The percentage shows how much a choice makes up of all choices made by the approach.

pared to the test set, mostly at the expense of external reorderings. This indicates that it is mainly when using internal reorderings that the SPTO approach differs from the SO approach.

As was expected from table 6.5, most of the instances where SPTO and SO differ, are of the I – E kind (64 instances). In these cases, where SPTO uses an internal and SO an external reordering to comply with the same rule, SPTO outperforms SO three times as much as SO outperforms SPTO. The reason for this must be that where SPTO is able to use the highly lexicalized, context-sensitive word choice of a phrase, SO uses individually translated fragments of the same area.

The other large group when comparing SPTO and SO, is the I – R group. Here, SO rejects a reordering that SPTO handles with an internal reordering. Almost 3/4 of these instances lead to a better translation by the SPTO system. Since SO is not able to find a satisfactory translation using the reordered path, it rejects the reordering. This means that it seeks the best translation based on the original source word order. This is exactly what the SPTO approach does here as well, and most of the time the SO approach will get the correct reordering from an internal reordering without knowing it. This is one of the reasons why the two approaches only yield different translations on a small subset of the sentences. Most often phrases are a very reliable source of information for local reordering. However, in the cases where they are not, and the reordered path does not help the SO approach, the SPTO approach has an advantage that leads to better translation.

The pattern is somewhat different compared to the no scoring and baseline approaches. These behave very similarly. In this relation, the SPTO ap-

proach only leads to better translation, when the other approaches choose to reject the proposed reordering. Compared to the no scoring approach, this clearly illustrates the SPTO scorings ability to motivate good reorderings, since the no scoring approach does not motivate in either direction, it simply chooses the most probable hypothesis based on translation and language model.

The perhaps most important result in table 6.6 is that the SPTO approach leads to better translation than the no scoring and baseline approaches by means of internal reordering in the I – R row. Either approach has used a phrase to cover the area, but via rule-based motivation, the SPTO has forced a less likely phrase with the correct word order through. This clearly shows that local reordering is not handled sufficiently by phrase internal reordering along. These need to be controlled too.

Finally, we observe that when the SPTO approach chooses to make an external reordering, the resulting translation is rarely worse than that made by the other approaches. The no scoring and baseline approaches are on the other hand reluctant to make use of external reordering (4 and 1 respectively). Instead, they rely very much on phrases to do their reordering. Table 6.5, however, revealed that the no scoring approach indeed often makes use of the external reordering option, so this must be due to different distribution of the diff set. One factor that may, however, play a role here, is the fact that the translation model is trained on unreordered source language only. It is likely that this brings a bias towards using the unreordered path in translation, since this is closest to training conditions. This would be especially evident in the no scoring approach, since the paths are not weighted otherwise.

Now, we will take a closer look at which rules lead to the reorderings. Table 6.8 shows the analysis from table 6.6 but here the comparison is based on the rules that have applied.

The table is sorted based on the improvement percentage of the rules. It is divided into four sections; 1) the top section contains rules where at least 50% of the rule applications lead to improvements, 2) contains rules where more applications lead to improvement than to decrease, 3) contains rules that neither improvement nor decrease, and 4) contain rules where more applications lead to decrease than to improvement.

Rule	Total	SO scoring			NO scoring			Baseline			Across systems	
	on test	Better	Same	Worse	Better	Same	Worse	Better	Same	Worse	Impr.	Decr.
26	7	1			1			1			100%	0%
6	106	4			3	1		4			92%	0%
7	500	3	3		6			4	2		72%	0%
1	96	6		1	4	1	2	4	1	2	67%	24%
3	711	5	4		6	3		6	3		63%	0%
5	133	3	2	1	3	3		3	2	1	50%	11%
4	407	3	4	1	4	4		4	4		46%	4%
2	719	7	7	1	5	9	1	4	8	3	36%	11%
9	349	5	2	1	2	5	1	1	7		33%	8%
21	1194	4	22	2	10	18		13	14	1	32%	4%
8	416	5	7	4	5	8	3	5	7	4	31%	23%
12	130		3			3			3		0%	0%
17	92		2			2			2		0%	0%
16	50		1			1			1		0%	0%
10	316		1			1			1		0%	0%
18	73			1		1				1	0%	67%
14	6			1			1			1	0%	100%
Total		46	58	13	49	60	8	49	55	13	41%	10%

Table 6.8: Manual analysis of rules used by the SPTO approach conducted on the 117 proposed reorderings in the 100 manually evaluated sentences. The translation of the reordered area is compared to other approaches as either better, same, or worse. The total column shows the total number of applications on the test set. Zero counts are excluded for ease of readability. The percentage of reorderings that result in improvement/decrease across the other systems is also reported. Rules are grouped based on their performance.

In general, very few of the rules seem to hurt performance, and only two rules in this sample hurt more than they help. Most notable are rules 14, 18, and 8. These rules handle the verb second phenomenon, stating that an NP and a finite verb should change place if they follow an "NP ," or a subordinate clause respectively. They are also very productive rules and bring substantial improvement compared to all other approaches, and even more important, they bring no decrease in translation quality.

Another interesting rule is number 21. This rule deals with the positioning of sentence adverbials in a subordinate clause. It is the most productive rule, but at the same time it seems to be very reliant in connection with the SPTO approach. It does not bring much improvement over SO scoring, but compared to the other approaches, it does very well, and most important, it hardly ever leads to a worse translation.

Reordering differences exemplified

Having looked at the overall statistics, we now present some reordering examples to illustrate properties of the system. First, we compare to the baseline system, then to the other pre-translation reordering approaches.

Table 6.9 contain two translations taken from the test set that display differences between the SPTO scoring and the baseline system. In translation 1), the subject (light shade) is correctly moved to the right of the finite verb (dark shade), which the baseline system fails to do. Moving the finite verb away from the infinite verb *'feature'*, however, leads to incorrect agreement between these. While the baseline correctly retains the infinite verb form (*'stå'*), the language model forces another finite verb form (the past tense *'stod'*) in the SPTO reordering approach.

Translation 2) illustrates the handling of adverbials. The first reordering is in a main clause, therefore, the adverbial is moved to the right of the finite verb. The second reordering occurs in a subordinate clause, and the adverbial is moved to the left of the finite verb. Neither of these are handled successfully by the baseline system, even though the reorderings are as local as they can be. This is because both sequences *'generally welcomes'* and *'aims principally'* were unknown to the phrase table.

In this case, the reordering leads to better word selection. The English *'aims to'* corresponds to the Danish *'sigter mod'*, which the SPTO approach

1	SRC	based on this viewpoint , every small port and every ferry port which handles a great deal of tourist traffic should feature on the european list .
	BL	baseret på dette synspunkt , ethvert lille havn og alle færgehavnen som håndterer en stor turist trafik skal stå på den europæiske liste .
	SPTO	baseret på dette synspunkt , skal alle de små havne , og alle færgehavnen som behandler mange af turister trafik stod på den europæiske liste .
2	SRC	the rapporteur generally welcomes the proposals in the commission white paper on this subject but is apprehensive of the possible implications of the reform , which aims principally to decentralise the implementation of competition rules .
	BL	ordføreren generelt bifalder forslagene i kommissionens hvidbog om dette emne , men er bekymret for de mulige konsekvenser af den reform , som sigter hovedsagelig at decentralisere gennemførelsen af konkurrencereglerne .
	SPTO	ordføreren bifalder generelt forslagene i kommissionens hvidbog om dette emne , men er bekymret for de mulige konsekvenser af den reform , som især sigter mod at decentralisere gennemførelsen af konkurrencereglerne .

Table 6.9: Examples comparing reorderings made by the SPTO approach and the baseline. SRC is source, and BL is baseline. The differently shaded elements have been reordered in the SPTO sentence.

gets correct. However, the baseline system translates *'to'* to its much more common translation *'at'*, because *'to'* is separated from *'aims'* by the adverbial *'principally'*.

Basically, both this and the previous agreement problems illustrate problems with the ngram language model. When words are separated from their governor, the ngram is not able to control a possible agreement between the two words due to its linear nature.

In table 6.10, we compare the SPTO scoring to the SO scoring and no scoring approaches. In the first translation, we see an example of why it is problematic that the SO scoring is ignorant to phrase internal reordering. All three translations produce the same reordering, which is the correct reordering here. In table 6.11, we show the phrase pairs used to translate first part of the sentence with the different approaches. The SO scoring has forced the use of the reordered path, since it is unaware of the internal reordering satisfying the reordering. The translation has therefore become more segmented, and this has lead to a much worse translation, since it did not catch the relation between *'believe'* and *'in'*. The fact that the no scoring system that does not weight the paths, got the correct translation shows that the SO scoring in fact hurts performance here.

The second translation of table 6.10 on the other hand shows an example where the motivation provided by the SPTO approach forces a good reordering through. Here, the no scoring approach chooses the most probable translation given no preference for either word order. By motivating the rule predicted word order, the SPTO approach finds another translation that is much better.

The SO scoring gets the same translation as the no scoring approach, since it rejects the rule. It cannot find a good translation using the reordered path, and it is unable to see the internal reordering utilized by the SPTO approach.

Problems with SPTO and possible solutions

Finally, we will look at some translations that proved problematic for the SPTO approach and discuss possible solutions. Table 6.12 shows two such cases. The first example shows a complex reordering, where the adverb *'fully'* should be moved inside the finite verb *'agree'*. The rule connected

1	SRC	i also believe in involving the ngos , mainstream society and the citizens of those countries .
	NO	jeg tror også på at inddrage ngo ’ erne , almindelige samfund og borgerne i disse lande .
	SO	jeg mener også med ngo ’ erne , almindelige samfund og borgerne i disse lande .
	SPTO	jeg tror også på at inddrage ngo ’ erne , almindelige samfund og borgerne i disse lande .
2	SRC	at present i feel there is a danger that if the proposal by the belgian government on these sanction mechanisms were to be implemented , we would be hitting first and examining only afterwards .
	NO	jeg i øjeblikket føler , der er fare for , at hvis forslaget fra den belgiske regering om disse sanktionsmekanismer blev gennemført , ville vi være rammer først og kun at undersøge bagefter .
	SO	jeg i øjeblikket føler , der er fare for , at hvis forslaget fra den belgiske regering om disse sanktionsmekanismer blevennemført , ville vi være rammer først og kun at undersøge bagefter .
	SPTO	i øjeblikket mener jeg , der er fare for , at hvis forslaget fra den belgiske regering om disse sanktionsmekanismer blevgennemført , ville vi være rammer først og kun at undersøge bagefter .

Table 6.10: Examples comparing reorderings made by the SPTO, no scoring, and SO scoring approaches. SRC is source. The differently shaded elements have been reordered in the SPTO sentence.

SO	i	believe	also	in	involving	the	ngos
	jeg	mener	også		med	ngo ’ erne	
SPTO/NO	i	also	believe	in	involving	the	ngos
	jeg	tror også på			at inddrage	ngo ’ erne	

Table 6.11: Phrase pairs used to translate example 1 in table 6.10

SRC	i	fully	agree
SO	jeg er helt		enig i
	1	*2*	*3*
SPTO	jeg	er helt enig	
	1	*3 2 3*	

Figure 6.4: Illustration of the translation of the first part of example 1 in table 6.12 showing the phrases used by the SO and SPTO approaches, and the phrase internal word alignment associated with the phrase.

with this example proposed to move *'fully'* to the right of *'agree'* and not inside it. Both approaches, however, get the correct reordering, but the SO scoring yields a better translation. The SPTO scoring fails to produce the *'i' (in)*, which makes the translation less acceptable.

Figure 6.4 shows the phrases used by the two approaches in translating the initial part of the sentence. The phrase that is used by the SO scoring, *'i fully → jeg er helt' (I is fully)*, has *'er' (is)* link to nothing, while the phrase used by the SPTO approach, *'fully agree → er helt enig' (is fully agreeing)*, links *'er' (is)* to *'agree'*, which is correct. That *'er' (is)* is linked to nothing in the first phrase, means that the reordering moving an *'is'* in between *'I'* and *'fully'* is invisible from the phrase internal word alignment.

The SPTO approach therefore scores the hypothesis that is used by the SO system, as not satisfying the rule, and the other hypothesis as satisfying the rule, even though none of the hypotheses in fact satisfy the rule. The fact that the second hypothesis is scored as satisfying the rule, even though not all of *'agree'* is moved to the left of *'fully'* as proposed by the rule, is a problem that we will discuss in connection with the next example. Here, we will focus on the general question of the quality of the phrase internal

1	SRC	i fully agree that the same rules concerning the quality of feedingstuffs have to apply to the quality of water consumed by animals .
	SO	jeg er helt enig i , at de samme regler om kvaliteten af foderstoffer skal gælde for kvaliteten af vand forbruges af dyr .
	SPTO	jeg er helt enig , at de samme regler om kvaliteten af foderstoffer skal gælde for kvaliteten af vand forbruges af dyr .
2	SRC	for me , the argument that nowadays very few transfers of euros are made to other countries in the euro area does not count either .
	SO	for mig er det argument , at i dag meget få overførsler af euro til andre lande i euroområdet ikke tælle hverken .
	SPTO	for mig , det argument , at i dag meget få overførsler af euro til andre lande i euroområdet tæller ikke , heller .

Table 6.12: Examples illustrating problems with the SPTO approach. SRC is source. The differently shaded elements have been reordered in the SO sentence.

word alignments used.

The phrase internal word alignment is the only additional resource used by the SPTO approach compared to the previous pre-translation reordering approaches. It is therefore also the only possible additional source of noise. For the SPTO approach to exhibit qualified evaluation of the relation between the source and the target sentence, it relies on the information from the phrase internal word alignments to be correct.

As was demonstrated in chapter 4, the word alignments produced by the GDF symmetrization of bi-directional IBM models are often full of error. In the English-Arabic experiments, the AER was at 22.99% for the baseline GDF alignment. A GDF alignment of the English-Danish CDEDT data also gets a rather high AER of 17.21%.

We do not expect the phrase internal word alignments to display such high error rates, since these have been exposed to the phrase extraction algorithm (see section 2.2.2), which will exclude sequence pairs that are not easily described. Excluding these unlikely alignment patterns such as wide covering spans will most likely filter out a lot of the bad alignment. Nonetheless, it may in fact be considered a hallmark of translation phrases that they often contain words that are not represented in their corresponding sequence, which may help translation but makes accurate word alignment impossible.

For translating the English-Danish test set a total of 168,429 phrase pair instances were used. The quality of these is difficult to evaluate, since we do not have a gold standard word alignment for Europarl, which the phrases are extracted from. And even if this were the case, an Alignment Error Rate would not necessarily provide a good reflection of the effect on translation quality, as reported by (Lopez & Resnik, 2006).

One parameter we can measure is the amount of unlinked words in the utilized phrase pairs. Of the 168,429 pairs used, 15,827 (9.4%) contain unlinked words. Danish, however, contains a lot more commas than English, so these should in fact often be unlinked, and it may therefore be a good idea to focus on alphanumeric words. 9,843 (5.8%) phrase pairs contained unlinked words with alphanumeric characters. As we will show in a moment, unlinked source words can be especially detrimental to the system, since these produce "holes" in the SPTO string. 9,317 (5.5%) phrase pairs

contained unlinked source words with alphanumeric characters.

Still, this does not provide an accurate measure for the effect this phenomenon has on translation quality. Some words are correctly not linked to anything, other words are incorrectly linked to something, and the wrongly unlinked words are not guaranteed to affect translation quality, if they do not appear in the reordering span.

We found no suitable means for isolating the effect of phrase internal word alignment errors without affecting other parts of the system. One experiment that may prove interesting, is the effect of using improved word alignment quality for the phrase internal alignments. This would not isolate the effect, but it may provide an indirect indication of its importance, since it would provide more accurate SPTO information. We leave these questions open for future experiments.

The second example in table 6.12 reveals that the SO approach produces a long reordering that the SPTO scoring does not. The idea of the reordering is good, but the word *'count'* should have been moved as well.

The reason that the SPTO approach does not produce this reordering as well, is that the SPTO string get very long. The decoder has to find a hypothesis that contains the SPTO string *'22 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21'*, because the 22nd word has moved in front of the fourth word. The probability that no other words should be reordered or aligned to nothing within these positions gets smaller as the SPTO string gets longer. In the example a phrase pair *'are made to → til' (to)* with *'til' (to)* correctly linking only to word 15 *'to'* is used. This would bring a hole in the SPTO, making it *'... 12 15 ...'*. The decoder did, however, find the desired SPTO string, but the hypothesis providing this string was too improbable, since it was very fragmented.

At this point, we return to figure 6.4. This illustrated that it is not enough to locate the desired SPTO in a hypothesis, in order for it to satisfy a rule. Even though, the SPTO *'3 2'* appears in the second hypothesis, the rule is not satisfied, since *'fully'* was in fact moved inside *'agree'*, and not to the right of it.

This problem speaks to the underlying idea of working with SPTO strings. A better solution is probably to think of them as sets. A way to do this is to say that the set of source word positions in the left sequence

may in the translation only appear to the right of the set of word positions in the right sequence. If a hypothesis upholds this, then the rule is satisfied.

Let us take the example from before. For this, the set-based SPTO algorithm would demand that the set of word positions in the left sequence *(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21)* may only appear to the right of the set of word positions in the right sequence *(22)*. By stating that they **may** appear, and not that they **should** appear, the holes mentioned before are treated correctly. If word positions *13* and *14* appear in the translation, they may only appear to the right of positions *22*, otherwise the rule is not satisfied. On the other hand, if they do not appear, this does not violate the rule. This way, the second hypothesis from figure 6.4 would also be scored in accordance with the rule, since word position *3* appears to the right of word position *2*.

It is, however, necessary to constrain the algorithm somewhat. The above would e.g. allow word position *(22)* to move all the way to the front of the sentence, and it would still be scored as satisfying the rule. This is done by demanding that no other word positions may appear inside the sequence made up of the union of the two sequence sets. This way the left and right sequences change place, but there are no demands for the word order inside the sequences.

We do not examine scoring based on SPTO sets any further in this thesis, but we are very interested in exploring this in future work. Instead, we now turn to the English-Arabic experiments.

6.4.2 English-Arabic SPTO experiment

Data

The system was trained on the same English-Arabic parallel corpus that was used to provide Giza++ alignments in chapter 4. This corpus consists of 126K sentences with 4.2M English and 3.3M Arabic words in the AR tokenization scheme. The domain is newswire (LDC-NEWS) taken from Arabic News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18).[2]

[2] All of the training data we use is available from the Linguistic Data Consortium (LDC): http://www.ldc.upenn.edu/.

Data	Sentences	English words	Arabic words
Training (TM)	126K	4.2M	3.3M
Training (LM)	5.5M	—	133M
Test MT04	5.4K	193K	144K
Test MT05	4.2K	143K	114K
Development	3.1K	102K	79K
Tuning	1.0K	34K	26K

Table 6.13: Statistics of data used for the English-Arabic experiments. TM is parallel data used for the translation model, and LM is monolingual data used for the language model.

The Arabic language model was trained on the 5.4M sentences (133M words) of newswire text in the 1994 to 1996 part of the Arabic Gigaword corpus.[2] We restricted ourselves to this part, since we were not able to run Pharaoh with a larger language model.

For test data, we used NIST MTEval test sets from 2004 (MT04) and 2005 (MT05)[3]. Since these data sets were created for Arabic-English evaluation with four English reference sentences for each Arabic sentence, we inverted the sets by concatenating all English sentences to one file. This means that the Arabic file contains four representations of each sentence. Following this merger, MT04 consists of 5.4K sentences with 193K English and 144K Arabic words, and MT05 consists of 4.2K sentences with 143K English and 114K Arabic words. MT04 is a mix of domains containing among other both speech, editorial and newswire. MT05 is on the other hand only newswire.

NIST MTEval test set from 2002 (MT02) was split in a tune set for optimizing decoder parameter weights and a development set for ongoing experimentation. The same merging procedure as for MT04 and MT05 was employed. This resulted in a tune set of 1.0K sentences with 34K English and 26K Arabic words, and a development set of 3.1K sentences with 102K English and 79K Arabic words.

[3] http://www.nist.gov/speech/tests/mt/

	Decoder choice	NO	SO	SPTO
MT04	Phrase internal (I)	4312 (21%)	127 (1%)	4417 (21%)
	Phrase external (E)	6265 (30%)	8951 (43%)	6891 (33%)
	Reject (R)	10249 (49%)	11774 (56%)	9499 (46%)
MT05	Phrase internal (I)	3660 (21%)	119 (1%)	3716 (22%)
	Phrase external (E)	5067 (29%)	7374 (43%)	5476 (32%)
	Reject (R)	8468 (49%)	9702 (56%)	8003 (47%)

Table 6.14: The reordering choices made based on the three pre-translation reordering approaches for the 20852 and 17195 reordering axes proposed by the rules for the MT04 and MT05 test sets. The percentage shows how much a choice makes up of all choices made by the approach on the given test set.

Results and discussion

The English-Arabic experiments are consistent with the experiments conducted for English-Danish. A lot of the findings are similar, and these similar findings will not be repeated here.

The SPTO reordering approach is evaluated on the MT04 and MT05 test sets. Results are listed in table 6.15 along with results on the development set. As for the English-Danish experiment, we also report on the *swap subset*. This set is, however, not as important here as it was before, since almost all sentences contain reorderings here as opposed to only 2/5 in the English-Danish experiment. We nevertheless report it for consistency. The MT04 swap set contains 5.1K sentences (93% of the entire set), and MT05 swap contains 4.0K sentences (95% of the entire set).

Table 6.14 shows the distribution of reordering choices made by the three scoring approaches for pre-translation reordering in decoding. Here, the pattern for the SO scoring from the English-Danish experiment (table 6.5 page 116) is even more outspoken. It hardly uses any phrase internal reorderings. Instead, it uses a lot more external reorderings than the other approaches. Compared to no scoring, the SPTO also leads to more external reorderings as was the case for English-Danish, but the difference is much smaller, and there is virtually no difference in internal reorderings.

Again we report on the baseline PSMT system both 1) without reordering restrictions, 2) a distortion limit of 4 (see section 2.2.2), and 3) no distortion

	System	Dev	MT04	MT04 Swap	MT05	MT05 Swap
BLEU	Pharaoh free	28.37	23.53	23.56	24.79	24.89
	Pharaoh dl4	29.52	24.72	24.77	25.88	25.99
	Pharaoh monotone	27.93	23.55	23.79	24.72	24.79
	no scoring	29.87	25.11	25.16	26.04	26.17
	SO scoring	29.84	25.06	25.11	26.01	26.13
	SPTO scoring	29.95	25.17	25.22	26.09	26.22
NIST	Pharaoh free	6.8311	6.4462	6.4492	6.7397	6.7487
	Pharaoh dl4	6.9610	6.5783	6.5856	6.8577	6.8670
	Pharaoh monotone	6.7539	6.4625	6.3971	6.6678	6.6746
	no scoring	6.9593	6.6215	6.6317	6.8268	6.8355
	SO scoring	6.9568	6.6135	6.6235	6.8233	6.8319
	SPTO scoring	6.9662	6.6251	6.6353	6.8310	6.8397

Table 6.15: Automatic evaluation scores for different systems. The SPTO system is significantly better than the light grey cells at a 95% confidence level.

allowed (monotone), 4) a system provided with a rule reordered word lattice but no scoring in the spirit of e.g. (Crego & Mariño, 2007), 5) the same system but with an SO scoring in the spirit of e.g. (Zhang et al., 2007b; Li et al., 2007), and finally 6) the same system but with the SPTO scoring. For the pre-translation reordering systems, we report on systems using the best performing rule set, which for English-Arabic is the one trained on GDF aligned data. As for the English-Danish investigation, we use the same parameter weights for the decoder. These are optimized on an SPTO scoring system with the IBMAC manual rules. For comparison, we optimize all systems, and test on the development set. This check reveals that the common parameters are not biased against any particular approach with only marginal +/- variation.

The results are much different for this distant language pair compared to the very close English-Danish. Here, a substantial gain is provided by the distortion reordering model, as long as it is restricted (here to a length of

	System	NIST	BLEU	Avr. human rating
MT04	Baseline	6.2800	24.07	3.0 (2.80)
	no scoring	6.4916	25.68	2.5 (2.43)
	SO scoring	6.4547	25.42	2.5 (2.64)
	SPTO scoring	6.5121	25.98	2.0 (2.13)
MT05	Baseline	6.5027	25.15	—
	no scoring	6.5291	26.29	—
	SO scoring	6.5132	26.02	—
	SPTO scoring	6.5513	26.49	—

Table 6.16: Evaluation on the diff sets. Average human ratings are medians with means in parenthesis, lower scores are better, 1 is the best score. The SPTO system is significantly better than the light grey cells.

4). This corresponds with previous findings for translation between Arabic and English (e.g. Koehn et al., 2005). If reordering is unrestricted, the gain from reordering is lost.

The SPTO approach outperforms the baseline system with 1.6% BLEU and 1.3% BLEU on the test sets ($p < 0.05$). Compared to the best performing Pharaoh system (dl4), brings an improvement of 0.5% BLEU and 0.1% BLEU respectively. The first of these results is significant ($p < 0.05$). As expected, the swap subsets give almost identical results as the entire sets, since they are almost identical. Most interesting about this set is that the SPTO system performs significantly better than the Pharaoh dl4 system on the MT05 data ($p < 0.05$).

We also look at the diff set in this experiment. This set focusses on the instances where the SO scoring and SPTO scoring lead to different translations. The diff set contains 767 for MT04 and 602 for MT05. Results are reported in table 6.16.

With exception of the no scoring approach on the MT05 data, the SPTO approach outperforms the other approaches significantly ($p < 0.05$). The systems are also manually evaluated. This is performed on 50 sentences from the MT04 test set by a single native speaker of Arabic. This shows the SPTO scoring to be significantly superior to the other methods ($p < 0.01$).

See section 2.4 for details on agreement and significance testing.

Analysis of reordering statistics

Same as for the English-Danish experiment, we list statistics for the reordering choices made by the SPTO system compared to the choices made by the other systems in table 6.17. See page 118 for details about notation and annotation process.

The table confirms that the SO scoring has a bias against phrase internal reorderings. Where the SPTO scoring leads to an internal reordering, the SO scoring most often prefers an external reordering. 25% of the times, this choice hurts performance.

The table also indicates that the SPTO approach is very strong in external reordering. Mainly compared to the baseline system, which is not surprising, since this performs best with restricted distortion, but also compared to the other systems.

We also here find that the SPTO approach has a positive effect on phrase selection. In the I – R row, the SPTO approach often leads to better translation where the no scoring and baseline systems choose a more likely phrase that does not contain the reordering.

Now, we will take a closer look at which rules lead to the reorderings. Table 6.18 shows the analysis from table 6.6 but here the comparison is based on the rules that have applied.

The table is sorted based on the improvement percentage of the rules. It is divided into four sections; 1) the top section contains rules where at least 50% of the rule applications lead to improvements, 2) contains rules where more applications lead to improvement than to decrease, 3) contains rules that neither improvement nor decrease, and 4) contain rules where more applications lead to decrease than to improvement.

Only few rules hurt performance, and the most productive rules lead to improvements more often then they lead to decrease. It is interesting that rule 11 often brings improvement over the SO scoring, while rule 4 often brings improvements over the baseline system. Rule 1, however, seems to bring consistent improvement over all the other approaches. In the following, we will concentrate on these three rules.

Reorder choice	SO scoring			NO scoring			Baseline			Across systems	
	Better	Same	Worse	Better	Same	Worse	Better	Same	Worse	Impr.	Decr.
E – E	3	28	1	1	27		1	4	3		
E – I							2	8		27%	5%
E – R	2	2		3	4	1	17	1			
I – E	7	19	2		1			1	2		
I – I		2			37			42		10%	3%
I – R		19		5	6		3	1			
R – E	5	8	1				1	20	3		
R – I					3		1	17	1	7%	5%
R – R	2	37	2		51	1	2	10			
Total	19	115	6	9	129	2	27	104	9	13%	4%

Table 6.17: Manual analysis of reordering choices made by the SPTO approach conducted on the 140 proposed reorderings in the 50 manually evaluated sentences. The choices are either E (external reordering), I (internal reordering), or R (rejected reordering). First choice (pre-dash) is that of the SPTO system, the second (post-dash) is of the other approach. For the baseline system, the choices are merely observations as to whether it satisfies the rules. It does not have this information in translation. The translation of the reordered area is compared to other approaches as either better, same, or worse. Zero counts are excluded for ease of readability. Percentage of SPTO reordering choices that result in improvement/decrease across the other systems is also reported.

Rule	Total on test	SO scoring			NO scoring			Baseline			Across systems	
		Better	Same	Worse	Better	Same	Worse	Better	Same	Worse	Impr.	Decr.
25	30	1				1		1			67%	0%
33	112		1			1		1			33%	0%
1	1437	4	9	1	4	9	1	5	9		31%	5%
11	2286	5	4	1	2	8		1	8	1	27%	7%
4	999	1	10		1	10		5	6		21%	0%
9	1482	2	7	1	1	9		2	8		17%	3%
5	1234	1	6		1	6		1	6		14%	0%
6	2324		17			17		6	11		12%	0%
32	1170	3	15			18		2	16		9%	0%
27	454		9			9		2	7		7%	0%
20	2574	2	15			16	1	1	15	1	6%	4%
35	121		2			2			2		0%	0%
39	12		1			1			1		0%	0%
12	510		2			2			2		0%	0%
8	59		1			1			1		0%	0%
7	482		1			1			1		0%	0%
19	91		3			3			3		0%	0%
23	979		4			4			4		0%	0%
24	918		5			5			3	2	0%	13%
18	229			1		1			1		0%	33%
22	38		1			1				1	0%	33%
21	321		1			1				1	0%	33%
2	776		1	1		2				2	0%	50%
36	141			1		1				1	0%	67%
Total		19	115	6	9	129	2	27	104	9	13%	4%

Table 6.18: Manual analysis of rules used by the SPTO approach conducted on the 140 proposed reorderings in the 50 manually evaluated sentences. The translation of the reordered area is compared to other approaches as either better, same, or worse. The total column shows the total number of applications on the entire test set. Zero counts are excluded for ease of readability. The percentage of reorderings that result in improvement/decrease across the other systems is also reported. Rules are grouped based on their performance.

Rule 11 is in fact a very liberal rule that states that a determiner and adjective can be moved to the right of anything, as long this is not followed by a noun. Since the SO scoring is not aware that a phrase contains the reordering, it will sometimes achieve a better score by doing a long move to another place in the sentence. Example (13) is one of the sentences where this happened. Here, the SPTO system moved *'the turkish'* after *'authorities'* with an internal reordering. The SO scoring, however, moved it after *'attacks'*, thereby creating a whole new meaning. This supports our claim that the SO scoring's inherent bias against internal reordering is a problem. At the same time, it illustrates that rule 11 in fact may not be a very good rule. In general, it may be a problem that either of the reordering sequences are allowed to be unspecified.

(13) the turkish authorities confirm they have dismantled the network responsible for the attacks and that six suspects are still abroad after having fled .

Rule 4 is a simple rule stating that a determiner and an adjective should move to the right of a following noun. This brings an improvement over the baseline in cases where an adjective modified noun phrase for which there exists no phrase table entry, has to be translated

The final rule we will discuss here, is rule 1. This rule is much more precise and detailed than the other two rules. It states that an adjective should move to the right of a following noun, if it is preceded by a determiner, and the noun is not followed by another noun or the preposition *of*. This rule is successful across systems, which indicates that this is not only a good, but also a general rule. We are unable to explain why this rule seems to fit better in an SPTO framework than with the other pre-translation reordering approaches, since it performs better both with internal and external reorderings.

6.4.3 Learning rules from different data

In this section, we will examine the effect of learning reordering rules under different settings. More precisely, we vary the domain of the texts, the size of the texts, and the method utilized to word align them. The rule sets

are evaluated by the effect they have on the quality of the translation as measured by the BLEU metric.

We learned rules from the data described in section 5.5. This means that for the English-Danish experiment, we employed training data from two domains; CDEDT, out of domain for the SMT system, and EP, same domain as the SMT system. We used two different word alignment methods; manual and GDF. This was only possible for the out-of-domain CDEDT data, since we do not have access to manually aligned EP. And finally, we varied the size of the in-domain data; 100K words (same size as the CDEDT), 300K words, and 772K words (upper limit for Ripper). For the English-Arabic experiment, we only varied the word alignment method on in-domain data of 176K words, but here we used three methods; manual, GDF, and combined (see chapter 4).

Table 6.19 shows results when the SPTO approach is provided with rule sets learned from different data sets. For ease of readability, we only report BLEU scores in this experiment. Only the significance relation between the best performing system within a group and the other systems is color coded. For the test and swap sets, the general picture between the other systems is the following; For DA, all systems are significantly better than no rules, and the CDEDT systems are significantly better than the EP systems. For MT04, all systems are significantly better than no rules, and for MT05, all differences are significant.

A very important result represented in the table is that no matter what rule set we employ, it helps translation. Compared to a monotone translation with no rule information, all the systems are significantly better, regardless of the language pair. This may not be surprising for the Arabic experiments, but the results for the Danish experiment should be seen in the light that the distortion-based model utilized by Pharaoh generally hurts performance. From this, we can draw the conclusion that the rule learning approach indeed provides useful rules.

A surprising finding is that increasing the learning data by up to 7.72 times has no effect. There is no significant difference when moving from 100K word of training data over 300K to 772K. On one hand, if the rules we learn are general and productive, then we should expect no gain from increasing the data set. Instead they should occur enough in the 2,427 re-

	Rule set	Dev	Test	Swap subset	Axes	Reorderings	External	internal
DA	CDEDT 100K manual	26.79	25.79	24.49	5715	4387 (77%)	2849 (50%)	1538 (27%)
	CDEDT 100K GDF	26.70	25.74	24.40	5335	3389 (64%)	2049 (38%)	1340 (25%)
	EP 100K GDF	26.68	25.67	24.26	3429	2040 (59%)	1172 (34%)	868 (25%)
	EP 300K GDF	26.68	25.67	24.25	3691	2095 (57%)	1217 (33%)	878 (24%)
	EP 772K GDF	26.68	25.65	24.21	3712	1916 (52%)	1081 (29%)	835 (22%)
	No rules	26.59	25.56	24.01	—	—	—	—
MT04	IBMAC manual	29.40	24.78	24.82	17657	8392 (48%)	5100 (29%)	3292 (19%)
	IBMAC combination	29.50	24.86	24.92	14870	7674 (52%)	4268 (29%)	3406 (23%)
	IBMAC GDF	29.95	25.17	25.22	20807	11308 (54%)	6891 (33%)	4417 (22%)
	No rules	28.24	23.77	23.79	—	—	—	—
MT05	IBMAC manual	29.40	25.93	26.04	14902	7203 (48%)	4236 (28%)	2967 (20%)
	IBMAC combination	29.50	25.61	25.73	11135	5668 (51%)	2947 (26%)	2721 (24%)
	IBMAC GDF	29.95	26.09	26.22	17182	9192 (53%)	5476 (32%)	3716 (22%)
	No rules	28.24	24.69	24.79	—	—	—	—

Table 6.19: BLEU scores when using the SPTO approach with rule sets learned from different data sets. DA is from the English-Danish experiment, and MT04 and MT05 are from the English-Arabic experiment. The best scoring system is significantly better than the light grey cells at a 95% level. We also report how many rule axes were predicted in the test set, how often this lead to a reordering, and how many of the axes lead to a phrase external/internal reordering.

orderings of EP 100K to be learned. On the other hand, we would expect that the 15,906 reorderings of EP 772K would provide basis for discovering better conditions for applying the rule. A plausible reason for the missing effect of scaling up the learning data may indeed lie with the choice of learning algorithm. Perhaps Ripper is not right for this task, considering the skewedness and size of the learning data.

Another unexpected result is that out-of-domain training data seem to provide better rules than in-domain data. We expect this to be a reflection of the quality of the parallel data. Where all sentences of the CDEDT corpus have been manually translated by the same professional translator, who was told to "make natural-sounding English translations that stayed as close to the Danish original as possible in terms of syntax" (Buch-Kromann et al., 2007). Sentences from Europarl are on the other hand often not direct translations. Instead they can be parallel translations of the same sentence in a third language. This means that they will often appear as very "loose" translations with variance in the content. They can for example have different foci, or entire passages may be missing in one language.

The fact that the out-of-domain data brings good performance, supports that the rules learned are general to the language pair, since they do not only function within the domain they are trained on. In addition, the results indicate the importance of precise translations for a delicate task such as learning reordering rules. We leave this question open for future experiments.

The final parameter we examine in this experiment, is the effect of utilizing different word alignment methods. Given its much higher accuracy, we expected manual alignments would lead to the best rules and thereby the best translations. This is, however, not the case for our English-Arabic experiments, where GDF rules lead to the best translations. It is difficult to say whether the combination alignment lies closer to manual alignment than the GDF alignment, and considering that the GDF rules outperform the manual rule, the question is whether this is a desirable property for this task. Either way, the rules learned from combination alignments do not lead to significant better results over any of the other systems.

A possible cause for this is that an SMT system thrives with a large search space. The larger the search space, the more translation hypotheses evalu-

	Rule set	Mean	Median
DA	CDEDT 100K manual	1.9	1
	CDEDT 100K GDF	1.9	1
	EP 100K GDF	1.4	1
	EP 300K GDF	1.4	1
	EP 772K GDF	1.4	1
MT04	IBMAC manual	2,076	10
	IBMAC combination	13,228	9
	IBMAC GDF	44,535,882,023	72
MT05	IBMAC manual	913	16
	IBMAC combination	2,899,852	10
	IBMAC GDF	17,856,158,928	120

Table 6.20: Statistics illustrating the search space provided by the pre-translation reordering approach. Values are average number of paths per word graph. We report both mean and median, since some sentences get an extremely large search space that makes the mean unreliable.

ated, and the more translation hypotheses, the higher likelihood of finding the one that fits the models best. This is for example seen when the beam in a beam-search decoder is lowered. While making the search space smaller, the risk of missing the most probable translation increases (Koehn, 2004a).

Table 6.20 illustrates the search space provided by the pre-translation reordering approach with different rule sets on different data. It shows the average number of paths per word graph created on the rule sets. It is clear to see that some rule sets provide the decoder with a lot more options than other rule sets. We therefore believe that this plays a major role in a rule set's effect on translation.

This assumption is supported when looking at the data in table 6.19 again. Figure 6.5 shows the correlation between the BLEU score and the number of axes suggested by a rule set. The correlation is almost perfect between the highly productive rule sets and their BLEU score. Even if we remove the somewhat artificial zero rule points, the R^2 values do not change much ($R^2_{DA} = 0.9924$, $R^2_{MT04} = 0.8603$, $R^2_{MT05} = 0.9317$). Considering the amount of measuring points, the correlation should of course only

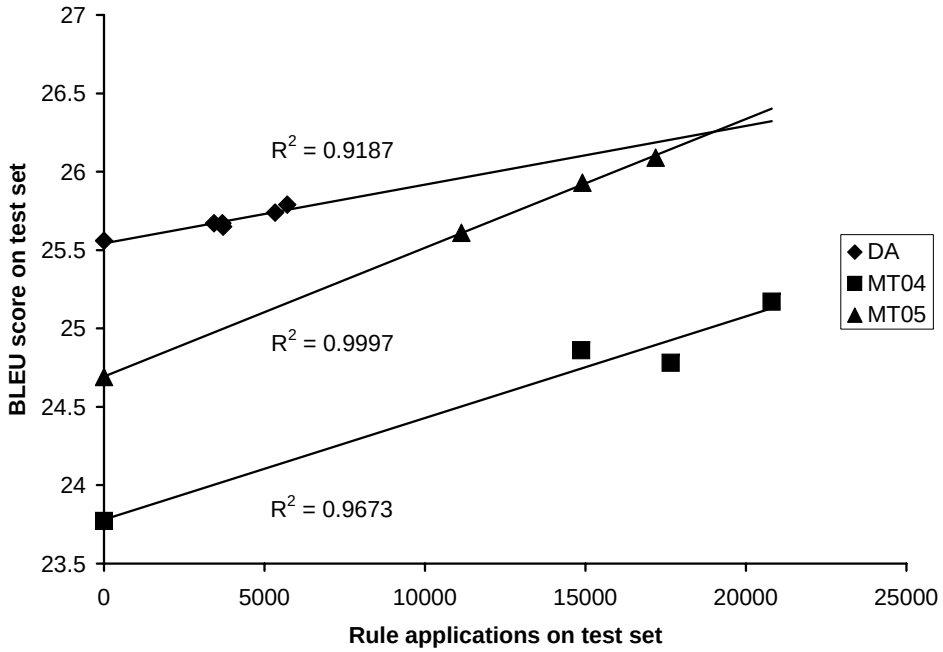


Figure 6.5: Illustration of the relation between the number of reordering axes a rule set assigns to the test sets, and the BLEU score an SPTO system gets with this rule set. DA the English-Danish test set, and MT04 and MT05 are the English Arabic test sets. The plot is annotated with a regression line and R^2 measure for correlation.

be taken as an indication.

In plain words, it seems that the pre-translation reordering approach carries a risk of narrowing down the search space too much. This means that the need for a larger search space gets very important, and the productive rule sets get the highest BLEU score, since they provide the largest search space. This should, however, be seen in the light that providing the linguistically motivated search space throughout our experiments has brought better performance than the much larger search space used by non-monotone Pharaoh.

Based on these findings, it is difficult to say whether the rules of a given set are better than those of another. It is therefore also not possible to conclude what effect the factors we have examined have on the quality of the rule. Some signs may indicate that the rule sets leading to higher BLEU

score contain better rules. Generally, a larger percentage of their suggested axes lead to reorderings as illustrated in table 6.19. This may suggest that the rules are better, since they are used. On the other hand, it may owe to the fact that more rules are applied at each axis, and the probability of one leading to a good word order is therefore greater. This is supported by the fact that more of their axes lead to external reorderings than is the case with the rule sets that lead to lower BLEU score. Since these can only be obtained through a reordered path, it indicates that the larger search space plays a role. The percentage of axes that lead to phrase internal reorderings is on the other hand fairly consistent between the rule sets.

We conclude that our approach was not balanced enough to examine more sensitive matters concerning the effect of varying the size, domain, and word alignment method on rule learning. We expect that more productive rules with higher recall at the expense of precision is needed for our approach. Much like the distortion model of traditional PSMT that for English-Arabic provides the best results when the relation between free reordering and no reordering is balanced off, our approach also needs to find this balance. In other words, we expect the correlation between BLEU score and application to wear off as precision gets too low.

6.5 Conclusion

We have described a novel approach to word reordering in SMT, which successfully integrates syntactically motivated reordering in phrase-based SMT. This is achieved by reordering the input string, but scoring on the output string. As opposed to previous approaches, this neither biases against phrase internal nor external reorderings. We achieve significant improvement in translation quality, measured by manual as well as automatic evaluations on an English-Danish and an English-Arabic task.

We also examine the effect that reordering rules learned under different circumstances have on translation. The rule sets were learned from different data set. We experiments on varying three parameters for these data; domain, size, and alignment method. We, however, found no consistent correlation between these parameters and translation quality. Instead, there was an almost perfect correlation with the number of times a rule

set applied to the test data. We believe this to be an indication that the rule sets in general are not productive enough for the SPTO approach in a pre-translation reordering framework, rather than that domain, size, and alignment method are unimportant factors in learning reordering rules.

Chapter 7

Conclusion

> The problem of MT is *only* one of quantity and capacity.
>
> (Bar-Hillel, 1965 [1955], p.183)

These opening words from Bar-Hillel were used to represent the idea that large masses of text can eliminate the need for linguistic knowledge in MT. Whether this turns out to be true as the amounts of electronic data avaliable increases dramatically, is impossible to foretell.

We believe that an SMT system can benefit from linguistic knowledge no matter how much data it has available. The ability to generalize at other levels provides a basis for better data exploitation. Some phenomena in fact seem impossible to handle without this knowledge.

Reordering is one such area where the worth of linguistic knowledge becomes apparent. Since the reordered elements most often pertain to the level of syntax, this information is an important part of their handling. In this thesis, we have explored this within a pre-translation reordering approach to phrase-based SMT. We first present and discuss the main results of the thesis followed by ideas on where this leads in the future.

7.1 Main results and discussion

We have introduced the novel SPTO approach to linguistically motivated reordering in phrase-based SMT. The SPTO approach operates by predicting reorderings between the source and the target sentence based on linguistic knowledge extracted from the source sentence. If a translation hypothesis

contains the sought reordering, it is motivated. The motivation assigned to a given reordering is decided by the probabilistic rule that suggested the reordering.

Whether a reordering occurs, is detected via the SPTO string. This is the target word order described in terms of origin source word positions. In order to keep track of this, phrase pairs are annotated with an internal word alignment. Since a reordering leads to a specific target word order, these SPTO strings can in turn be used to represent a reordering. The decoder can therefore check whether the SPTO string of a translation hypothesis contains the SPTO string associated with a given reordering.

In our experiments, we employ multiple levels of linguistic knowledge from word form to syntactic information. The approach is, however, not restricted to these. It can exploit any level of linguistic knowledge that might be helpful in reordering. The only linguistic concept it is tied to, is the word form. We regard it as a strength of this approach that it is neither tied to translation phrase units nor syntactic phrase units.

The approach was tested in the context of a pre-translation reordering framwork, since this provides an interesting setting for rule-based reorderings in PSMT. The approach is, however, not restricted to this framework. On an English-Danish and an English-Arabic task, the SPTO approach performs significantly better than both previous pre-translation reordering approaches and a state-of-the-art PSMT system. A result that is backed by human evaluations.

An important question concerning the SPTO approach is the influence of the phrase internal alignments. Since these provide the source position origins, the SPTO approach is at the mercy of their information. Bad word alignment may corrupt the approach with noise. This is an intervening factor that the previous approaches described are not exposed to. In our experiments, this potential source of noise did not outweigh the advantages of the approach.

The rules used by the SPTO approach are automatically learned based on a rich set of linguistic information. The approach yields a relatively small rule set of predominantly general rules. The rule sets proved their worth as a reordering information source for the SPTO approach.

We also examined the effect of varying domain, size, and alignment

method for the data that the rules were learned from. We found no significant correlation between these and translation quality. Instead, their productivity showed high correlation with translation quality. We believe this indicates that these rule sets are not productive enough for the SPTO approach in a pre-translation reordering framework, rather than that domain, size, and alignment method are unimportant factors in learning reordering rules.

Finally, we provide a new approach for improving automatic word alignment. Our approach learns from hand aligned data how to combine several automatic word alignments to one superior word alignment. The automatic word alignments are created from the same data that has been preprocessed with different tokenization schemes. Thus utilizing the different strengths that different tokenization schemes exhibit in word alignment. We achieve a 38% error reduction for the automatic word alignment. Utilizing the improved alignments for learning reordering rules did, however, not bring an improvement as mentioned above.

7.2 Future directions

The work done in this thesis has brought interesting insights and answered a lot of questions. These insights and answers have, however, raised new questions. In this section, we will describe some of the paths we wish to investigate in the future.

SPTO sets instead of strings

We realize that specifying desired reorderings as SPTO strings may be too restrictive. When doing long reorderings, the desired SPTO string has stringent requirements to the area that is crossed. If this area contains unlinked source words or other reorderings, then these requirements are not fulfilled. This means that a good reordering may be scored as bad, due to irrelevant circumstances.

In future work, we want to examine the idea of SPTO sets. Here, the reordering sequences are defined as sets of words that should change places. The words in the left set may only appear to the right of the other set in the translation. This way, it is not required that all source words should

be translated, nor that the words within the reordered sequences should appear in a specific order.

We believe the high requirements of the SPTO string approach has crippled long distance reordering in our experiments. SPTO sets are therefore expected to bring a lot more long distance reordering. Our only concern about a set-based approach is that it may exercise too little restriction on reordering. We, however, do not expect this to be the case.

Beyond pre-translation reordering

We are interested in examining the SPTO scoring approach outside a pre-translation reordering framework. An intersting experiment would be to simply replace the distortion model in a traditional PSMT system with the SPTO scoring. This would provide less restricted external reordering. Where the pre-translation reordering approach can only make an external reordering if the reordered source order transfers to the target side, this approach could use any source order that would lead to the desired target order.

This experiment would require that the SPTO scoring would be implemented in a decoder based on another search algorithm. An obvious possibility here would be to integrate it in the open-source beam-search decoder, Moses (Koehn et al., 2007).

More productive rule extraction

In our experiments, we developed a rule learning approach that was aimed at learning the most general rules. We succeded in doing so, but the rules were not as productive as hoped for. For the experiment mentioned above in *Beyond pre-translation reordering*, we expect we would need much more productive rules.

This should also be seen in light of the almost perfect correlation between BLEU score and the number of suggested reordering axes. One explanation for this correlation may indeed be that the lacking rule productivity has restricted phrase external reordering too much. By having more reordering axes, the search space is extended, and the decoder therefore has more hypotheses to choose from. It is therefore difficult to tell whether the rules

of the best performing set are better, or they merely provide the decoder with more options. This clearly needs more investigation, since we would expect the correlation to wear off at some point.

One factor that we expect plays a major role in connection with the lacking productivity, is that on this very skewed distribution Ripper is only able to learn less than half the reorderings available. In the word alignment experiments, we were able to reduce this factor dramatically via heuristic means with hardly any loss. This was not possible for the rule learning, which resulted in more difficult circumstances for Ripper. For the word alignment learning, 1/5 of the features were positive. For the rule learning, the ratio was around 1/500 for English-Danish and 1/50 for English Arabic.

Following these comments, we are very interested in exploring a Maximum Entropy approach in line with (Li et al., 2007). This is likely to be more productive and result in a wider set of reordering possibilities.

The effect of phrase internal alignment on SPTO

The SPTO approach brings the notion of phrase internal alignments into the picture. This is represented in the traditional PSMT system with the lexical weights, but with the SPTO approach, their accuracy becomes an important issue. We were not able to answer the question on the influence of the noise brought by this potential noise factor. In future work, we are very interested in designing an experiment that isolates this factor within our approach without affecting other parts of the system.

One experiment that may prove interesting, is the effect of using improved word alignment quality for the phrase internal alignments. This would not isolate the effect, but it may provide an indirect indication of its importance, since it would provide more accurate SPTO information.

Another interesting extension is to let word link outside the phrase. If a target phrase contains a word that does not link to anything, one option would be to look for possible linking candidates in neighboring phrases.

Phrase extraction from weighted source

Finally, we still believe the pre-translation reordering approach is an interesting framework for controlling reordering. One aspect that was lacking in our approach was training on reordered source language. This provides the system with better odds, since it will be translating the language it has been trained on, when translating a reordered path. In our experiments, on the other hand, the decoder is only provided with information from original source language, whether it is translating the original or a reordered path. We expect this will produce a slight bias towards the original path. This may for example partly explain why the unweighted no scoring system makes very few phrase external reorderings.

For these reasons, we are very interested in exploring a comparable approach for non-deterministic pre-translation reordering. Zhang et al. (2007b) propose such an approach, but the reordered source is not based on their rules. Instead, it is based on an unfolded word alignment.

We are more interested in training on reordered source language based on our rule set. More precisely, we want to find a way of extracting phrase pairs from a weighted word lattice that is aligned to a string. This way, we would obtain an environment that is consistent with the translation situation.

One way to do this would this, would be to generate a word alignment between the original word order and the target. Then a word graph is created for each source sentece based on the reordering rules, and each node is annotated with its link to the target sentence. All consistent phrase pairs are then extracted while traversing the paths.

Appendix A

List of reordering rules

This appendix contains all the rule sets learned as described in chapter 5. Rules are ordered as learned by Ripper. Some rules display a lot of redundancy. When discussing rules in the thesis, redundancy was excluded for ease of readability. A sequences like "MAIN/SUB" means that the first part of the sequence is in a main clause, while the last part is in a subordinate clause. "FVF" is a category consisting of the finite verb forms: "AUX", "MD", "VBD", "VBP", and "VBZ";

Rule learned based on CDEDT (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1	PS: <s> PP	PS: , NP	POS: FVF		83%
2	WORD: , PS: <s> PP ,	PS: NP ! WORD: ”	POS: FVF		83%
3	WORD: , PS: SBAR ,	PS: NP ! WORD: ”	POS: FVF		82%
4	WORD: , PS: ADVP , ! WORD: however ,	PS: NP ! POS: NNP	POS: FVF		70%
5	PS: ADVP ! WORD: however	PS: , NP ! WORD: , ”	POS: FVF		70%
6	! PS: S PS: NP	PS: , NP ! WORD: , ” SUB: MAIN	POS: FVF		66%
7	WORD: , PS: NP ,	PS: NP ! WORD: ”	POS: FVF		55%
8	SUB: MAIN PS: <s> NP	PS: ADVP	POS: FVF		62%
9	POS: RB ! WORD: <s> ”	PS: NP	POS: FVF		74%
10	PS: NP PS: PP	PS: NP SUB: MAIN	POS: FVF		51%
11	WORD: , ”	PS: NP	POS: FVF		93%
12	WORD: if WORD: <s> if	PS: NP	POS: FVF		53%
13	WORD: , PS: S ,	PS: NP ! WORD: ”	POS: FVF		40%

Rule learned based on CDEDT (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
14	SUB: MAIN	PS: NP POS: PRP	POS: FVF	! WORD: T/L POS: IN	87%
15	WORD: ,	SUB: MAIN PS: NP ADVP	POS: FVF		88%
16	SUB: MAIN ! POS: IN	PS: NP POS: PRP	POS: FVF PS: VP		47%
17	PS: CC NP	PS: ADVP	POS: FVF		55%
18	PS: NP	PS: NP SUB: MAIN	POS: FVF	! SUB: MAIN/SUB	36%
19	WORD: ,	PS: NP POS: PRP	POS: FVF TO		72%
20		PS: , NP ! WORD: , ”	POS: FVF	POS: IN	52%
21		PS: FVF	POS: RB	SUB: SUB PS: VP	71%
22	WORD: , PS: <s> PP ,	PS: NP	POS: FVF VBG		86%
23	WORD: ,	PS: NP	POS: FVF VBN		56%
24	PS: NP CC	PS: ADVP	POS: FVF		63%
25	! POS: IN ! WORD: <s>	PS: NP	POS: FVF	POS: RB . </s>	50%
26	WORD: ,	PS: NP POS: PRP	! WORD: T/L POS: VB	PS: NP	83%
27		WORD: dkk		! POS: CD ! POS: FVF	97%

Rule learned based on CDEDT (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1	WORD: ,	POS: PRP	POS: FVF		68%
2	WORD: ,	PS: NP ! WORD: ”	POS: FVF		44%
3	! PS: S PS: <s> PP	PS: , NP	POS: FVF		77%
4	WORD: , ”	POS: PRP	POS: FVF		91%
5	PS: <s> NP	PS: ADVP	POS: FVF		60%
6	PS: ADVP POS: CC RB		POS: FVF		60%
7		PS: FVF	POS: RB	SUB: SUB PS: VP	62%
8	PS: RB	POS: PRP	POS: FVF		52%
9		POS: FVF	POS: RB PS: ADVP	SUB: SUB	44%
10	! PS: VP ! PS: INTJ	POS: , PRP	POS: FVF		58%
11	PS: ADVP WORD: now	SUB: MAIN	POS: FVF		75%
12	! WORD: , ! WORD: with sage at	WORD: dkk		! WORD: million	95%
13	PS: CC NP	PS: ADVP	POS: FVF		53%
14	! PS: VP	PS: , NP WORD: , there	POS: FVF		62%

Rule learned based on CDEDT (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
15	POS: CC	POS: RB	POS: FVF	SUB: MAIN	60%
16	! WORD: <s> PS: NP	SUB: MAIN POS: PRP	POS: FVF	WORD: T/L	59%
17	SUB: MAIN WORD: then	PS: NP	POS: FVF		70%
18	SUB: MAIN POS: .	PS: NP POS: PRP	POS: FVF		83%
19	WORD: , the	SUB: MAIN POS: NN	! WORD: T/L POS: FVF		83%

Rule learned based on EP 100K (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1	WORD: , SUB: SUB/MAIN	POS: PRP	POS: FVF		75%
2	WORD: , ! POS: <s> NNP NN ,	POS: PRP	POS: FVF WORD: would	SUB: MAIN/SUB	61%
3	WORD: , PS: PP , ! POS: IN NN ,	POS: PRP	POS: FVF	! PS: NP ! POS: VB TO VB ! PS: IN NP FVF	55%

Rule learned based on EP 100K (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
4	SUB: MAIN ! POS: CC PS: PP	POS: PRP	POS: FVF		50%
5	WORD: , PS: SBAR ,	PS: NP	POS: FVF		48%
6	WORD: , PS: PP , PS: NN ,	POS: PRP	POS: FVF	SUB: MAIN/SUB	66%
7	PS: <s> NP	SUB: MAIN PS: ADVP	POS: FVF		58%
8	SUB: MAIN POS: RB WORD: then	POS: PRP	POS: FVF	! SUB: MAIN/SUB	85%
9	SUB: MAIN PS: <s> ADVP ,	POS: PRP	PS: FVF		35%
10	WORD: T/L PS: S SUB: SUB	POS: PRP	POS: FVF		73%
11	SUB: MAIN ! POS: CC ! POS: NN , ! POS: RB ,	POS: PRP WORD: i	POS: FVF	! SUB: SUB/MAIN	50%
12		POS: FVF SUB: SUB	POS: RB	PS: VP	59%
13	! POS: NNP NN PS: NN	SUB: MAIN POS: , PRP	POS: FVF		43%
14	SUB: MAIN PS: NP	PS: NP WORD: all	PS: FVF		71%

Rule learned based on EP 300K (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1	WORD: , WORD: T/L SUB: SUB/MAIN	POS: PRP	POS: FVF		68%
2	WORD: , PS: PP ,	POS: PRP	POS: FVF	SUB: MAIN/SUB	52%
3	SUB: MAIN ! POS: CC PS: PP , WORD: T/L ! POS: NN ,	POS: PRP	POS: FVF		57%
4	! WORD: <s> ! POS: CC PS: NP	SUB: MAIN POS: PRP	POS: FVF		53%
5	PS: <s> NP	SUB: MAIN PS: ADVP	POS: FVF		60%
6	SUB: MAIN PS: SBAR , PS: PP ,	PS: NP	POS: FVF		58%
7	SUB: MAIN WORD: , ! WORD: president , WORD: finally ,	PS: NP	POS: FVF		58%
8	WORD: , WORD: T/L PS: SBAR ,	PS: NP	POS: FVF		37%

Rule learned based on EP 300K (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
9	SUB: MAIN PS: ADVP	PS: NP	POS: FVF		43%
10	SUB: MAIN ! POS: CC ! POS: <s> NNP NN , ! WORD: <s> however , ! WORD: T/L	POS: PRP SUB: MAIN WORD: i	POS: FVF	PS: VP	46%
11	WORD: , WORD: T/L ! PS: S , SUB: MAIN	PS: NP POS: DT NN	PS: FVF		76%
12	WORD: , ! WORD: president ,	POS: PRP	POS: FVF WORD: have		42%
13		POS: FVF SUB: SUB	POS: RB	PS: VP	56%
14	WORD: , PS: PP , POS: IN DT NN ,	POS: PRP	POS: FVF		41%

Rule learned based on EP 300K (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
15	PS: <s> PP	SUB: MAIN POS: , PRP	POS: FVF		47%
16	WORD: , ! WORD: president , ! PS: <s> NP ,	PS: NP POS: PRP POS: PRP	POS: FVF	POS: RB SUB: MAIN/SUB	50%
17	SUB: MAIN WORD: , PS: PP , POS: NN NN ,	PS: NP	POS: FVF		57%
18	WORD: , PS: <s> PP , POS: NN ,	PS: NP WORD: there	POS: FVF		77%
19	WORD: ,	POS: PRP ! WORD: i	POS: FVF	POS: VBN	60%
20	SUB: MAIN PS: NP PS: <s> PP	PS: NP	POS: FVF	WORD: a	75%

Rule learned based on EP 772K (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1	WORD: , PS: SBAR ,	POS: PRP	POS: FVF		68%
2	SUB: MAIN PS: PP ,	POS: PRP	POS: FVF		46%
3	PS: <s> NP	SUB: MAIN PS: ADVP ! WORD: cannot	POS: FVF		59%
4	SUB: MAIN WORD: , PS: SBAR ,	PS: NP	POS: FVF		50%
5	SUB: MAIN PS: PP	PS: NP	POS: FVF		48%
6	SUB: MAIN ! POS: CC ! POS: NN , ! WORD: <s> however ,	POS: PRP	POS: FVF	SUB: MAIN PS: VP . POS: VB	49%
7	SUB: MAIN ! POS: CC ! POS: <s> NNP NN , POS: RB	POS: PRP	POS: FVF	SUB: MAIN	49%
8	WORD: , PS: PP ,	PS: NP WORD: there	POS: FVF	SUB: MAIN	52%
9		POS: FVF SUB: SUB	POS: RB	PS: VP	52%

Rule learned based on EP 772K (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
10	WORD: , ! POS: <s> NNP NN , PS: <s> PP , ! PS: NN ,	PS: NP WORD: the commission	POS: FVF		68%
11	WORD: , ! POS: <s> NNP NN , ! WORD: however , POS: RB ,	PS: NP POS: PRP	POS: FVF		39%
12	WORD: , PS: PP , ! PS: , PP , POS: NNS , POS: NN NNS ,	PS: NP	POS: FVF		61%
13	! PS: IN ! WORD: <s> PS: NP PS: VP POS: NN	PS: NP SUB: MAIN	POS: FVF		79%
14	SUB: MAIN ! POS: IN ! POS: CC POS: RB	PS: NP	POS: FVF	! PS: VP SUB: SUB/MAIN	52%
15	WORD: , PS: PP , ! PS: , PP ,	PS: NP	POS: FVF	PS: ADVP VP	44%

Rule learned based on IBMAC (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1	! POS: JJ	PS: JJ	POS: NN WORD: president		90%
2	POS: IN SUB: SUB	PS: JJ	POS: NNS	! WORD: T/L	83%
3	POS: IN	PS: JJ ! WORD: ”	POS: NNS	! WORD: ,	69%
4	! POS: JJ ! PS: NP	PS: JJ ! WORD: ”	POS: NN	! WORD: , ! SUB: MAIN	56%
5	! POS: JJ ! PS: NP	PS: JJ	POS: NN	POS: IN ! WORD: of	65%
6	! POS: JJ	PS: JJ	POS: NNS	PS: PP WORD: in	66%
7	POS: DT WORD: the	PS: JJ ! WORD: ”		PS: NN PS: NN VP ! POS: NN NN	73%
8	POS: DT	PS: JJ ! WORD: ” ! WORD: sharm	POS: JJ	PS: NN ! POS: NN NN	68%
9	POS: CD	PS: JJ	! WORD: T/L POS: NNS		66%
10	POS: TO VB	PS: JJ	! WORD: T/L POS: NNS		81%
11	POS: IN	PS: JJ	! WORD: T/L POS: NN		46%

Rule learned based on IBMAC (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
12	POS: DT	PS: JJ	! WORD: T/L	PS: NNS	49%
13	! POS: JJ	PS: JJ ! WORD: ”	! WORD: T/L POS: NNS	! WORD: , ! POS: FVF	49%
14		PS: JJ WORD: last	! WORD: T/L	WORD: . </s>	96%
15		PS: JJ WORD: israeli	! WORD: T/L		49%
16	WORD: <s>		! WORD: T/L POS: FVF	PS: SBAR	68%
17	WORD: <s>	PS: JJ	! WORD: T/L	PS: SBAR .	97%
18	POS: DT	PS: JJ	! WORD: T/L POS: NN POS	! POS: JJ NN	90%
19	POS: IN ! WORD: in ! WORD: T/L	PS: JJ ! WORD: ”	! WORD: T/L	POS: IN SUB: SUB/MAIN	62%
20	WORD: ”	PS: JJ ! WORD: many	! WORD: T/L POS: NNS ! WORD: ”		87%
21	WORD: <s>	PS: JJ ! WORD: urgent	! WORD: T/L	PS: VP </s> PS: FVF	73%
22		PS: JJ WORD: last	! WORD: T/L	! POS: CD	57%
23	! POS: JJ ! PS: NP ! WORD: the ! WORD: T/L	PS: JJ ! WORD: ”	! WORD: T/L PS: NN	! PS: NN ! SUB: MAIN/SUB	43%

Rule learned based on IBMAC (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
24	PS: DT		! WORD: T/L PS: NN	POS: NN ! PS: NN NN	51%
25	POS: IN ! WORD: in	PS: JJ ! WORD: ”	! WORD: T/L ! POS: CC JJ	SUB: SUB/MAIN ! WORD: T/L	61%
26		WORD: .	! WORD: T/L WORD: ”		94%
27	WORD: <s>	PS: NP	! WORD: T/L POS: FVF		43%
28		PS: JJ WORD: palestinian	! WORD: T/L		40%
29	PS: NNP		! WORD: T/L	WORD: (afp)	74%
30	PS: IN	POS: NN	! WORD: T/L PS: NN		42%
31	POS: IN ! WORD: in	PS: JJ ! WORD: ”	! WORD: T/L	WORD: .	61%
32	POS: IN ! WORD: in	PS: JJ ! WORD: ”	! WORD: T/L	PS: VP . </s>	47%
33	POS: IN ! WORD: in	PS: JJ ! WORD: ”	! WORD: T/L POS: JJ NN		58%
34		POS: JJ JJ	! WORD: T/L PS: NN		46%
35		PS: JJ ! WORD: ” WORD: next	! WORD: T/L		67%

Rule learned based on IBMAC (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
36		PS: JJ	! WORD: T/L POS: NNS ,		48%
37	! POS: JJ ! PS: NN	POS: NN	! WORD: T/L PS: NNS ! WORD: ”	POS: IN	71%
38		POS: NNP WORD: gaza	! WORD: T/L PS: NN		75%
39		WORD: united	! WORD: T/L PS: NNS		76%
40	POS: DT	POS: JJ ! WORD: ” ! WORD: international	! WORD: T/L POS: JJ NN	PS: NN	56%
41		POS: IN JJ	! WORD: T/L PS: NN		35%
42	WORD: <s> NNP POS	! POS:	! WORD: T/L PS: NN	WORD: :	77%
43		! POS: JJ	! WORD: T/L PS: NN WORD: minister		50%
44		! PS: IN NNP	! WORD: T/L PS: NN WORD: evening	! WORD: ,	67%
45	PS: DT WORD: the		! WORD: T/L PS: NN	POS: NNS	63%

Rule learned based on IBMAC (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
46	! PS: NP ! POS: JJ	PS: JJ ! WORD: ”	! WORD: T/L POS: NN NN	! WORD: ,	40%
47	! PS: NP ! POS: JJ	PS: JJ ! WORD: ”	! WORD: T/L POS: JJ NNS	PS: PP	79%
48	! PS: NP ! POS: JJ	PS: JJ ! WORD: ”	! WORD: T/L POS: NN NNS		47%
49	! PS: NP ! POS: JJ ! POS: RB ! WORD: ,	PS: JJ ! WORD: ”	! WORD: T/L POS: NN ,	! PS: SBAR	55%
50		POS: VBG	! WORD: T/L PS: NN	WORD: in	66%
51	POS: IN WORD: of	POS: NN	! WORD: T/L PS: NNS ! WORD: ”		72%
52	POS: IN	! POS: CD POS: JJ CC JJ	! WORD: T/L PS: NNS		90%
53	POS: IN ! WORD: by	! POS: CD POS: NN	! WORD: T/L PS: NNS ! WORD: ”		59%
54	POS: CD) :	PS: NP	! WORD: T/L POS: FVF		81%
55		! WORD: T/L ! POS: CD POS: IN JJ	! WORD: T/L PS: NNS		43%

Rule learned based on IBMAC (manual)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
56	! PS: NN		! WORD: T/L PS: NN WORD: month	WORD: .	84%
57	POS: DT SUB: MAIN	POS: JJ	! WORD: T/L POS: NNS POS	POS: NN	84%
58	WORD: the international	POS: JJ	! WORD: T/L PS: NN		72%
59	POS: DT WORD: all ! WORD: of all	POS: JJ ! WORD: ”	! WORD: T/L	! SUB: SUB/MAIN	90%
60	PS: NP IN ! WORD: in	! WORD: T/L	! WORD: T/L PS: NN	PS: PP ! WORD: of WORD: in ! POS: IN DT	63%
61		! WORD: T/L POS: JJ JJ	! WORD: T/L PS: NNS	WORD: .	88%

Rule learned based on IBMAC (combination)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1		PS: JJ	POS: NNS		49%
2	POS: IN	PS: JJ	POS: NN		56%
3	! POS: JJ ! SUB: MAIN	PS: JJ	PS: NN		47%
4	PS: DT PS: NP IN DT		PS: NN	! PS: NN ! WORD: ,	49%
5	PS: DT POS: IN DT	PS: JJ JJ	PS: NN	! WORD: ,	45%
6	! POS: JJ ! PS: NP	PS: JJ ! WORD: ”	! WORD: T/L PS: NN	! PS: NN ! WORD: , ! SUB: MAIN/SUB ! POS: NNS	50%
7	POS: DT POS: IN DT	PS: JJ ! WORD: ”	! WORD: T/L		37%
8	PS: IN WORD: of	! POS: DT POS: DT JJ	PS: NN	! POS: NN	60%
9	POS: DT	PS: JJ ! WORD: ”	! WORD: T/L	POS: IN	49%
10		PS: JJ WORD: last	! WORD: T/L		59%

Rule learned based on IBMAC (combination)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
11	PS: IN WORD: of POS: NNS IN	! POS: DT ! WORD: T/L POS: NN	PS: NN		77%
12	PS: IN	! POS: DT ! WORD: T/L PS: JJ JJ	PS: NN		62%
13	PS: IN	! POS: DT ! WORD: T/L	PS: NN	WORD: . </s>	43%
14	POS: DT		PS: NN	PS: PP ! PS: IN NP	37%
15	PS: IN	! POS: DT ! WORD: T/L	PS: NN	WORD: in ! POS: IN DT	57%
16		POS: DT JJ	! WORD: T/L PS: NN	! PS: NN	36%
17		POS: IN JJ	! WORD: T/L PS: NN		43%
18	PS: IN WORD: of	! POS: DT ! WORD: T/L PS: DT JJ NN	! WORD: T/L PS: NN	! WORD: ,	68%
19		POS: NNP POS JJ	! WORD: T/L PS: NN WORD: minister	! WORD: ,	93%
20	PS: IN	! POS: DT ! WORD: T/L POS: DT JJ JJ	! WORD: T/L PS: NN	PS: PP	76%

Rule learned based on IBMAC (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
1	POS: DT	PS: JJ	POS: NN	! PS: NN ! WORD: of	68%
2	POS: DT	PS: JJ ! WORD: ”		! PS: NN SUB: SUB	59%
3	POS: DT	PS: JJ ! WORD: ” ! WORD: first		! PS: NN ! WORD: , ! WORD: of	55%
4		POS: DT JJ	PS: NN		58%
5	PS: DT PS: IN DT	! PS: T/L	PS: NN	! PS: NN ! WORD: , ! WORD: of	57%
6		PS: JJ	POS: NNS		51%
7	PS: DT	! PS: T/L	PS: NN	! PS: NN ! WORD: , ! WORD: of POS: IN	64%
8	! POS: JJ	POS: JJ	PS: NN WORD: president		93%
9	! POS: JJ	PS: JJ	PS: NN	! PS: NN ! WORD: , ! WORD: of	52%

Rule learned based on IBMAC (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
10	POS: IN	! POS: DT PS: DT NN	PS: NN	! POS: NN	70%
11		PS: DT JJ		! PS: NN	53%
12	POS: IN	! POS: DT ! PS: NP IN DT ! WORD: T/L ! POS: IN DT ! POS: VBG DT	PS: NN	POS: .	64%
13	POS: IN	! POS: DT ! PS: NP IN DT POS: DT JJ JJ	PS: NN		59%
14	PS: DT	! POS: DT ! PS: T/L SUB: SUB POS: JJ JJ	PS: NN		53%
15		! POS: DT POS: IN DT JJ	PS: NN		55%
16	POS: IN	! POS: DT ! PS: NP IN DT PS: DT JJ NN	PS: NN	PS: VP . </s>	77%
17	PS: DT SUB: SUB	! POS: DT ! PS: T/L POS: NN	PS: NN		51%

Rule learned based on IBMAC (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
18	PS: DT	! PS: T/L	PS: NN ! WORD: ” ! WORD: ”	PS: VP ! SUB: MAIN/SUB	40%
19	! POS: JJ	PS: JJ ! WORD: ” WORD: last	! WORD: T/L		57%
20	PS: DT		! WORD: T/L PS: NNS		43%
21	POS: IN PS: NP IN	! POS: DT ! PS: NP IN DT ! WORD: T/L	PS: NN	PS: VP	42%
22	POS: IN	! POS: DT ! PS: NP IN DT ! WORD: T/L	PS: NN	WORD: </s>	62%
23	POS: IN	! POS: DT ! PS: NP IN DT ! WORD: T/L	! WORD: T/L PS: NN	! WORD: of PS: PP	46%
24	! POS: JJ POS: IN ! WORD: in	PS: JJ ! WORD: ”	! WORD: T/L		39%
25	PS: DT	! POS: DT ! PS: T/L POS: JJ	! WORD: T/L PS: NN	! WORD: T/L	41%
26		! POS: DT POS: IN JJ	! WORD: T/L PS: NN		50%

Rule learned based on IBMAC (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
27		WORD: .	! WORD: T/L	WORD: </s>	96%
28	! SUB: MAIN	! POS: DT	! WORD: T/L PS: NN	! WORD: of ! SUB: SUB WORD: :	84%
29	PS: DT	! POS: DT ! PS: T/L POS: JJ NN	! WORD: T/L PS: NN	! WORD: of ! WORD: , ! PS: VP . ! PS: NN	63%
30	! SUB: MAIN	! POS: DT PS: DT NN	! WORD: T/L PS: NN	! WORD: of	48%
31			! WORD: T/L PS: NNS WORD: states		61%
32	WORD: <s>		! WORD: T/L POS: FVF	POS: IN	68%
33	! WORD: for	! POS: DT ! POS: IN DT POS: JJ JJ	! WORD: T/L PS: NN ! WORD: afp	! WORD: of ! PS: NN ! SUB: SUB ! WORD: , ! POS: IN ! SUB: SUB/MAIN	70%

Rule learned based on IBMAC (GDF)					
No	**LC**	**LS**	**RS**	**RC**	**Prob.**
34	POS: DT	PS: JJ ! WORD: ”	! WORD: T/L	SUB: SUB	36%
35		POS: IN JJ	! WORD: T/L PS: NNS		61%
36	! POS: JJ ! WORD: ’s	! POS: DT ! POS: IN DT ! POS: POS JJ	! WORD: T/L PS: NN ! WORD: afp WORD: minister	! WORD: of ! WORD: , ! POS: FVF ! POS: IN	92%
37		! POS: DT	! WORD: T/L PS: NN	! WORD: of POS: IN WORD: in POS: IN NNP PS: PP FVF NP	70%
38	WORD: to	! POS: DT ! POS: IN DT PS: DT JJ NN	! WORD: T/L PS: NN ! WORD: afp	! WORD: of	88%
39	WORD: <s>	PS: JJ ! WORD: ”	! WORD: T/L	SUB: SUB	92%

References

Yaser Al-Onaizan & Kishore Papineni (2006). Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'06)* (pp. 529–536). Sydney, Australia.

Necip F. Ayan (2005). *Combining linguistic and machine learning techniques for word alignment improvement*. PhD thesis, University of Maryland, College Park.

Roger Bakeman & John M. Gottman (1997). *Observing interaction: an introduction to sequential analysis*. Cambridge, MA, USA: Cambridge University Press.

Satanjeev Banerjee & Alon Lavie (2005). 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* (pp. 65–72). Ann Arbor, Michigan: Association for Computational Linguistics.

Yehoshua Bar-Hillel (1965 [1955]). Idioms. In William N. Locke & A. Donald Booth (Eds.), *Machine translation of languages* (3rd Ed.). The MIT Press.

Adam L. Berger, Vincent J. Della Pietra & Stephen A. Della Pietra (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, *22(1)*, 39–71.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, John D. Lafferty, Robert L. Mercer & Paul S. Roossin (1990). A sta-

tistical approach to machine translation. *Computational Linguistics*, *16(2)*, 79–85.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra & Robert L. Mercer (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, *19(2)*, 263–311.

Matthias Buch-Kromann, Jürgen Wedekind & Jakob Elming (2007). The Copenhagen Danish-English dependency treebank v. 2.0. http://www.isv.cbs.dk/∼mbk/cdt2.0.

Tim Buckwalter (2004). Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02, ISBN 1-58563-324-0.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz & Josh Schroeder (2007). (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 136–158). Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne & Philipp Koehn (2006). Re-evaluating the role of bleu in machine translation research. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*. Association for Computational Linguistics.

Eugene Charniak (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)* (pp. 132–139).

Stanley F. Chen & Joshua T. Goodman (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, USA.

Colin Cherry & Dekang Lin (2006). Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'06)* (pp. 105–112). Sydney, Australia.

David Chiang (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 263–270). Morristown, NJ, USA: Association for Computational Linguistics.

David Chiang (2007). Hierarchical phrase-based translation. *Computational Linguistics*, *33(2)*, 201–228.

William Cohen (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning* (pp. 115–123).

William Cohen (1996). Learning trees and rules with set-valued features. In *Fourteenth Conference of the American Association of Artificial Intelligence*.

Michael Collins, Philipp Koehn & Ivona Kucerova (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 531–540).

Bernard Comrie (1989). *Language universals and linguistic typology* (2nd Ed.). Cambridge, MA, USA: Blackwell Publishers.

Josep M. Crego & José B. Mariño (2007). Syntax-enhanced n-gram-based smt. In *Proceedings of the 11th Machine Translation Summit (MT Summit XI)* (pp. 111–118).

Arthur E. Dempster, Nan M. Laird & Donald B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, *39(B)*, 1–38.

Mona Diab & Nizar Habash (2006). Tutorial: Arabic dialect processing. http://www1.ccls.columbia.edu/∼cadim/presentations.html.

Philip Diderichsen & Jakob Elming (2005). A corpus-based approach to topic in danish dialog. In *Proceedings of the ACL Student Research Workshop* (pp. 109–114).

George Doddington (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceeding of the ARPA Workshop on Human Language Technolog*.

Bonnie J. Dorr, Lisa Pearl, Rebecca Hwa & Nizar Habash (2002). DUSTer: A method for unraveling cross-Language divergences for statistical word-Level alignment. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas (AMTA'02)*. Tiburon, California.

Jakob Elming (2005). Sdmt alignment manual. unpublished.

Jakob Elming (2006). Transformation-based corrections of rule-based mt. In *Proceedings of the 11th annual conference of the European Association of Machine Translation (EAMT'06)*.

Jakob Elming (2008). Syntactic reordering integrated with phrase-based smt. In *Proceedings of the ACL Workshop on Syntax and Structure in Statistical Translation (SSST-2)*.

Jakob Elming (accepted for publication). Syntactic reordering integrated with phrase-based smt. In *Proceedings of the 22th International Conference on Computational Linguistics (COLING'08)*.

Jakob Elming & Nizar Habash (2007). Combination of statistical word alignments based on multiple preprocessing schemes. In *Proceedings of the 8th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL'07)* (pp. 25–28).

Jakob Elming, Nizar Habash & Josep M. Crego (to appear). Combination of statistical word alignments based on multiple preprocessing schemes. In Cyril Goutte, Nicola Cancedda, Marc Dymetman & George Foster (Eds.), *Learning Machine Translation*. The MIT Press.

Heidi J. Fox (2002). Phrasal cohesion and statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*.

Alexander Fraser & Daniel Marcu (2006). Semi-supervised training for statistical word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'06)* (pp. 769–776). Sydney, Australia.

Andrew T. Freeman, Sherri L. Condon & Christopher M. Ackerman (2006). Cross linguistic name matching in english and arabic. In *Proceedings of the 7th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL'06)* (pp. 471–478). New York City, USA.

Michel Galley, Mark Hopkins, Kevin Knight & Daniel Marcu (2004). What's in a translation rule? In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL'04)* (pp. 273–280). Boston, Massachusetts, USA: Association for Computational Linguistics.

Jesús Giménez & LLuís Màrquez (2007). Linguistic features for automatic evaluation of heterogeneous mt systems. In *Proceeding of the ACL Workshop on Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics.

Sharon Goldwater & David McClosky (2005). Improving statistical MT through morphological analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP'05)* (pp. 676–683). Vancouver, Canada.

Nizar Habash (2007a). Arabic morphological representations for machine translation. In A. van den Bosch & A. Soudi (Eds.), *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Nizar Habash (2007b). Syntactic preprocessing for statistical machine translation. In *Proceedings of the 11th Machine Translation Summit (MT Summit XI)* (pp. 215–222).

Nizar Habash & Owen Rambow (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 573–580). Ann Arbor, Michigan.

Nizar Habash & Fatiha Sadat (2006). Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 7th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL'06)* (pp. 49–52). New York, NY.

Nizar Habash, Abdelhadi Soudi & Tim Buckwalter (2007). On Arabic transliteration. In A. van den Bosch & A. Soudi (Eds.), *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

W. John Hutchins & Harold L. Somers (1992). *An introduction to machine translation*. London: Academic Press.

Abraham Ittycheriah & Salim Roukos (2005). A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP'05)* (pp. 89–96). Vancouver, British Columbia, Canada.

Frederick Jelinek (1998). *Statistical methods for speech recognition*. The MIT Press.

Douglas Jones, Edward Gibson, Wade Shen, Neil Granoien, Martha Herzog, Douglas Reynolds & Clifford Weinstein (2005). Measuring human readability of machine generated text: three case studies in speech recognition and machine translation. In *Proceedings of 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 5 (pp. 1009–1012).

Daniel Jurafsky & James H. Martin (2006). *Speech and language processing* (2nd Ed.). Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Kevin Knight (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, *25(4)*, 607–615.

Philipp Koehn (2004a). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA'04)* (pp. 115–124).

Philipp Koehn (2004b). Statistical significance tests for machine translation evaluation.

Philipp Koehn (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit (MT Summit X)*.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne & David Talbot (2005). Edinburgh system description for the 2005 iwslt speech translation evaluation. In *International Workshop on Spoken Language Translation 2005 (IWSLT'05)*.

Philipp Koehn & Hieu Hoang (2007). Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (pp. 868–876).

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin & Evan Herbst (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions* (pp. 177–180). Prague, Czech Republic.

Philipp Koehn & Christof Monz (2006). Manual and automatic evaluation of machine translation between european languages. In *Proceedings on the Workshop on Statistical Machine Translation* (pp. 102–121). New York City.

Philipp Koehn, Franz J. Och & Daniel Marcu (2003). Statistical phrase-based translation. In *Proceedings of the 4th Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'03)* (pp. 48–54).

Klaus Krippendorff (1980). *Content analysis: an introduction to its methodology*. Sage Publications.

J. Richard Landis & Gary G. Koch (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*, 159–174.

Alon Lavie & Abhaya Agarwal (2007). 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 at the 45th Meeting of the Association for Computational Linguistics (ACL'07)* (pp. 228–231). Prague, Czech Republic.

LDC (2005). Linguistic data annotation specification: Assessment of fluency and adequacy in translations. revision 1.5. http://www.ldc.upenn.edu/Projects/TIDES/Translation/TransAssess04.pdf.

Young-Suk Lee (2004). Morphological analysis for statistical machine translation. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL'04)* (pp. 57–60). Boston, USA.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou & Yi Guan (2007). A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45rd Annual Meeting of the Association for Computational Linguistics (ACL'07)* (pp. 720–727). Prague, Czech Republic.

Yang Liu, Qun Liu & Shouxun Lin (2005). Log-linear models for word alignment. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 459–466). Ann Arbor, Michigan.

Adam Lopez & Philip Resnik (2006). Word-based alignment, phrase-based translation: what's the link? In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA'06)* (pp. 90–99).

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow & Dalila Tabessi (2006). Developing and using a pilot dialectal arabic treebank. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy.

Mohamed Maamouri, Ann Bies, Tim Buckwalter & Wigdan Mekki (2004). The Penn Arabic Treebank : Building a large-scale annotated Arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools* (pp. 102–109). Cairo, Egypt.

Christopher D. Manning & Hinrich Schütze (1999). *Foundations of statistical natural language processing*. Cambridge, MA, USA: The MIT Press.

Daniel Marcu, Wei Wang, Abdessamad Echihabi & Kevin Knight (2006). SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)* (pp. 44–52). Sydney, Australia.

Mitchell P. Marcus, Beatrice Santorini & Mary A. Marcinkiewicz (1994). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, *19(2)*, 313–330.

Tom M. Mitchell (1997). *Machine learning*. McGraw-Hill.

Robert C. Moore (2005). A discriminative framework for bilingual word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP'05)* (pp. 81–88).

Makoto A. Nagao (1984). A framework of a mechanical translation between japanese and english by analogy principle. In A. Elithorn & R. Banerji (Eds.), *Artificial and human intelligence* (pp. 173–180). Amsterdam, North-Holland.

John A. Nelder & Roger Mead (1965). A simplex method for function minimization. *The Computer Journal*, *7*, 308–313.

Sonja Nießen & Hermann Ney (2004). Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, *30(2)*.

Franz J. Och (2003). Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL'03)* (pp. 160–167). Sapporo, Japan.

Franz J. Och & Hermann Ney (2000). Improved statistical alignment models. In *Proceedings of the 38th Meeting of the Association for Computational Linguistics (ACL'00)* (pp. 440–447).

Franz J. Och & Hermann Ney (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL'02)* (pp. 295–302).

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

Franz J. Och & Hermann Ney (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, *30(4)*, 417–449.

Franz J. Och, Christoph Tillmann & Hermann Ney (1999). Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP'99)* (pp. 20–28). University of Maryland, College Park, MD, USA.

William O'Grady, Michael Dobrovolsky & Francis Katamba (1997). *Contemporary linguistics: an introduction* (Adapted Ed.). London, England: Longman.

Kishore Papineni, Salim Roukos, Todd Ward & Wei-Jing Zhu (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL'02)* (pp. 311–318). Philadelphia, PA.

Maja Popović & Hermann Ney (2004). Towards the use of word stems and suffixes for statistical machine translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)* (pp. 1585–1588). Lisbon, Portugal.

Adwait Ratnaparkhi (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1996 Conference on Empirical Methods in Natural Language Processing (EMNLP'96)* (pp. 133–142). Somerset, New Jersey.

Fatiha Sadat & Nizar Habash (2006). Combination of arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'06)* (pp. 1–8). Sydney, Australia.

Satoshi Sato & Makoto A. Nagao (1990). Toward memory-based translation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING'90)* (pp. 247–252).

Sidney Siegel (1956). *Nonparametric statistics for the behavioral sciences*. New York, USA: McGraw-Hill.

Harold L. Somers (2003). An overview of ebmt. In M. Carl & A. Way (Eds.), *Recent advances in example-based machine translation*, Volume 21 of

Text, Speech and Language Technology chapter 1, (pp. 3–57). Kluwer Academic Publishers.

Andreas Stolcke (2002). Srilm – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

Eiichiro Sumita, Hitoshi Iida & Hideo Kohyama (1990). Translating with examples: a new approach to machine translation. In *Proceedings of the third conference on theoretical and methodological issues in machine translation of natural language (TMI)*.

Ben Taskar, Simon Lacoste-Julien & Dan Klein (2005). A discriminative matching approach to word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP'05)* (pp. 73–80).

Christoph Tillmann (2001). *Word re-ordering and dynamic programming based search algorithm for statistical machine translation*. PhD thesis, RWTH Aachen University, Aachen, Germany.

Christoph Tillmann, Stefan Vogel, Hermann Ney & Alex Zubiaga (1997). A DP-based search using monotone alignments in statistical translation. In *Proceedings of the 35th Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL'97)* (pp. 289–296). Somerset, New Jersey.

Russell S. Tomlin (1995). Focal attention, voice, and word order: and experimental, cross-linguistic study. In Pamela Downing & Michael Noonan (Eds.), *Word order in discourse* (pp. 517–554). Amsterdam/Philadelphia: John Benjamins Publishing Company.

Davide Turcato & Fred Popowich (2003). What is example-based machine translation? In M. Carl & A. Way (Eds.), *Recent advances in example-based machine translation*, Volume 21 of *Text, Speech and Language Technology* chapter 2, (pp. 59–81). Kluwer Academic Publishers.

Bernard Vauquois (2003). Automatic translation—a survey of different approaches. In S. Nirenburg, H. Somers & Y. Wilks (Eds.), *Readings in machine translation* (pp. 333–337). Cambridge, MA, USA: The MIT Press.

Chao Wang, Michael Collins & Philipp Koehn (2007). 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'07)* (pp. 737–745).

Warren Weaver (2003 [1949]). Translation. In S. Nirenburg, H. Somers & Y. Wilks (Eds.), *Readings in machine translation* (pp. 13–17). Cambridge, MA, USA: The MIT Press.

Fei Xia & Michael McCord (2004). Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)* (pp. 508–514). COLING.

David Yarowsky, Grace Ngai & Richard Wicentowski (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of First International Conference on Human Language Technology Research (HLT'01)* (pp. 161–168).

Richard Zens, Franz J. Och & Hermann Ney (2002). Phrase-based statistical machine translation. In *Proceedings of the 25th Annual German Conference on AI (KI'02)* (pp. 18–32).

Ying Zhang, Stephan Vogel & Alex Waibel (2004). Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.

Yuqi Zhang, Richard Zens & Hermann Ney (2007a). Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation* (pp. 1–8). Rochester, New York: Association for Computational Linguistics.

Yuqi Zhang, Richard Zens & Hermann Ney (2007b). Improved chunk-level reordering for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)* (pp. 21–28). Trento, Italy.

Liang Zhou, Chin-Yew Lin & Eduard Hovy (2006). Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)* (pp. 77–84). Sydney, Australia: Association for Computational Linguistics.

Simon Zwarts & Mark Dras (2007). Syntax-based word reordering in phrase-based statistical machine translation: why does it work? In *Proceedings of the 11th Machine Translation Summit (MT Summit XI)* (pp. 559–566).