University of Zurich
Institute of Computational Linguistics

HS 2009

# Syntactically Enriched Statistical Machine Translation from English to German

Lizenziatsarbeit der Philosophischen Fakultät der Universität Zürich
Referent: Prof. Martin Volk

Rico Sennrich
Obere Engelaustrasse 3
9010 St.Gallen
rico.sennrich@gmx.ch

September 2009

# Abstract

Statistical Machine Translation from English to German is challenging due to the morphological richness of German and word order differences between the two languages. Aiming at a better translation of selected linguistic phenomena, we explore the use of automatically computed syntactic information in translation models, language models, and for reordering. We find that the syntactic head and role of source text words are helpful during translation, even though data in the syntactically enriched translation models is sparse. We investigate the effects of data sparseness in the syntactically enriched models, and how a deterioration in translation quality can be avoided by combining them with more general models. By combining syntactically enriched translation models and a surface form translation model through a back-off-chain, we obtain a small improvement in translation quality.

# Acknowledgments

I would like to express my gratitude to all the people who supported me while I was writing this thesis. First, I would like to thank my supervisor Prof. Martin Volk, whose encouragement, support and constructive feedback I very much appreciated.

This thesis was partly inspired by working with Gerold Schneider on syntactic parsing. I am greatly indebted to him for helping me become familiar with the Pro3Gres parser, which allowed me to integrate it neatly into the SMT system.

I am grateful to Christian Hardmeier, whose experience with Moses helped me plan this thesis and find the resources I needed.

My thanks go to Helena Müller and Moira Kindlimann, who competently and carefully proofread this thesis and showed me where I could improve its clarity.

I would like to thank the countless people who not only made the large number of digital resources I used for this thesis freely available, from various linguistic applications and SMT tools up to the corpus data, but whose careful documentation also allowed me to quickly dive into the world of SMT and focus on my actual research question.

My final thanks go to my family, who made my studies possible.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| LM | Language model |
| MERT | Minimum Error Rate Training |
| MLE | Maximum Likelihood Estimation |
| MT | Machine Translation |
| NP | Noun phrase |
| POS | Part-of-speech |
| SMT | Statistical Machine Translation |
| STTS | Stuttgart-Tübingen Tagset |
| TM | Translation model (not translation memory) |
| WSD | Word sense disambiguation |

# 1 Introduction

## 1.1 Machine Translation

We live in a world in which most physical boundaries to communication have been overcome. Thanks to a dense net of data wires and radio technology, communication between any two points on Earth has become almost instantaneous. Just as importantly, the World Wide Web has allowed us to organize information comprehensively, so that we can find and access what is relevant to us in spite of the sheer amount of data available. However, one important boundary remains: the language barrier.

Even before the advent of the Internet, it had been recognized that the steadily increasing volume of publications and the growth of international cooperation was putting huge demands on the translation community (Hutchins 1978). Today, we are regularly confronted with situations in which a translation of a text, spoken or written, would be desirable, but too expensive and/or too slow to be economically viable. Machine Translation (MT) tries to close this gap by automating the translation process, and as a result providing fast translations at little cost.

The automatic translation from one language to another is an extremely challenging task, mainly due to the fact that natural languages are ambiguous, context-dependent and ever-evolving. In this light, being able to automatically translate any type of text from and to any language is an unrealistic goal. Still, MT has proven useful, and Hutchins (1999) lists various areas where MT systems have been successfully employed. MT systems can quickly produce rough translations of electronic texts, and are popular as a reading assistance for foreign-language Web pages and e-mails. Another field in which MT systems can excel is the real-time translation of repetitive language, for

instance weather reports. Generally, MT mostly fills niches where human translations would be too costly or slow. If the quality of the translations needs to be high and the texts are linguistically sophisticated, human translators are unrivalled. However, it is possible to use MT systems as a supporting tool in the human translation process. Depending on the degree of automation, these approaches are called *Machine-Assisted Human Translation* or *Human-Assisted Machine Translation* (Kay 1982).

Finding real-world applications for MT systems is easy. More problematic is the question as to how automatic translations can be achieved. The various approaches to MT can be divided into two major categories: rule-based MT and corpus-based MT.

> The difference between [rule-based MT] and [corpus-based] MT can be described as "deductive" vs. "inductive" MT. The fundamental difference between these is the source of knowledge that eventually determines the behavior of the system. Deductive MT systems rely on linguists and language engineers, who create or modify sets of rules in accordance with their knowledge, expertise, and intuition. In inductive MT systems, the rules are derived by the system itself and rely on a given set of translation examples. (Carl et al. 2000)

Both approaches have their strengths and weaknesses. The biggest disadvantage of rule-based systems is the amount of human work required to write the rules, and the fact that the rules can rarely be re-used for other tasks. The degree to which a rule-based system can be re-used depends on its level of abstraction. A rule-based system that uses little or no abstraction, but executes rules that translate directly from one surface string into another, called *direct translation*, needs a new set of rules for every translation direction. In other words, one needs to write $n^2 - n$ sets of rules if one wishes to cover all translation directions for $n$ languages. The ultimate goal of research in rule-based MT would be a language-independent representation level, or *interlingua*, which could serve as an intermediate step in every translation task. This would reduce the number of rule sets that need to be explicitly written to $2n$ for $n$ languages: one set for every

source language that analyses a text and creates its interlingua representation, and one for every target language that generates the output sentence from the interlingua. An interlingua that can capture the meaning of a sentence without ambiguities and loss of information is still far away, though.[1]

Corpus-based approaches also depend on human work, but this work mainly consists of building multilingual corpora containing thousands of sentences and their translations. If such corpora already exist, an MT system can be built with relatively little work. The component that learns translation rules from the corpus is mostly language-independent, and can thus be re-used for different translation tasks. Koehn (2005) demonstrates the ease with which MT systems can be built with a corpus-based approach by training 110 translation systems – from and to each of 11 languages – within three weeks.

## 1.2 Objectives and Structure

In this thesis, we will work with Statistical Machine Translation (SMT), a corpus-based approach that uses mathematical models instead of linguistically motivated ones. The quality of SMT heavily depends on the language pair and translation direction. Generally, translations between related languages are easiest. Koehn (2005) reports good results for translations between the Romance languages French, Spanish and Italian. In contrast, translations into German are unsatisfactory, even when translating from related languages such as Dutch and English. In Koehn's study, only systems translating into Finnish performed worse.

What German and Finnish have in common is their rich morphology, which has been identified as the main difficulty. German noun phrases are marked for case, gender and number, which are expressed as different word endings. The statistical model often has no means of generating the right word ending from the source word, especially if the

[1] Naturally, direct and interlingual translation are not the only possible levels of abstraction, but just the extremes.

source text has no case markings. Our main research question is if we can resolve translation ambiguities by incorporating automatically computed syntactic information into the translation process. We will work with the translation pair English – German and a corpus of proceedings of the European Parliament. Additionally, we will investigate other methods of integrating syntactic information into the baseline SMT system.

The following chapter introduces the theoretical foundations of Statistical Machine Translation, with a focus on the Moses system, which we used for all experiments. Then, we will describe how translation quality is measured and how we can try to optimize the results with a given model. The subsequent chapter is devoted to a linguistic discussion as to what information a syntactic analysis provides and how we can use it for SMT. This is followed by the experimental section, where we describe and evaluate a baseline system and different experiments we performed. As a conclusion, we will summarize the most important findings and provide an outlook for possible future research.

# 2 Statistical Machine Translation

## 2.1 Probabilistic Foundations

The basic idea in Statistical Machine Translation (SMT) is to assign probabilities to translations. If we task a hundred professional translators with translating a simple sentence such as example 1, there are several translations that we expect (e.g. example 2), while we would be surprised by others (such as example 3).

1. All the world's a stage.

2. Die ganze Welt ist eine Bühne.

3. Etwas ist faul im Staate Dänemark.

Intuitively, we expect a translation to be good, assuming that the translators are competent. In other words, we deem a good translation to be more probable than a bad one. A probabilistic approach such as SMT exploits this connection between the probability of a translation and its quality. Instead of trying to produce good translations, SMT aims to produce probable ones; quality is achieved indirectly. Whether a translation is probable or not is, as in other corpus-based approaches, learned from a *parallel corpus*, a collection of sentences and their respective translations. For now, let us suppose that our corpus was built by asking a hundred people to translate example 1.

Formally, we express the probability of an event $x$, denoted by $P(x)$, on a scale from 0 (impossible) to 1 (certain). The probability of a target sentence $T$ being the translation of a source sentence $S$ is expressed as $P(T|S)$. This is a *conditional probability*, which says that we can disregard all sentences that are not translations of $S$. Probabilities are

estimated by counting from a set of data, a parallel corpus in our case. Hence, $P(T|S)$ is estimated by counting the number of sentence pairs where $T$ is the translation of $S$, and dividing the result by the total frequency of $S$ (equation 1).

$$P(T|S) \approx \frac{count(T,S)}{count(S)} \tag{1}$$

This approximation is known as *maximum-likelihood estimation* (MLE), since it leads to probabilities that maximize the likelihood of the data we observe in the corpus. Going back to the one hundred translations of *all the world's a stage*, choosing the best translation is easy: we simply select the most frequent one. This will only result in a bad translation if the majority of the people mistranslated the sentence; in those cases, nobody expects an automatic system to succeed anyway.

Fundamentally, all statistical approaches in natural language processing can be divided into two steps: firstly, train a statistical model by estimating probabilities from a training corpus; secondly, use the model to generate or find an output $O$ that maximizes $P(O|I)$ for a given input $I$. The first step is known as *training phase*, while the second is the *decoding phase*, owing to Weaver's famous metaphor of translations as a cryptographical problem (1949/1955). Statistical approaches are successfully used in many fields of computational linguistics, including parsing, tagging, and speech recognition.[1]

So far, our discussion of SMT has been largely hypothetical. In reality, we do not have the luxury of possessing a hundred human translations for every sentence we wish our SMT system to translate. Actual SMT systems do not operate on the sentence level due to limited data. To show what is realistically possible, we will now describe how the data for SMT is obtained, and then proceed to introduce actual, more fine-grained SMT systems.

[1] see (Manning and Schütze 1999) for an overview.

## 2.2 Parallel Corpora

Statistical Machine Translation is corpus-based, and consequently requires a parallel corpus to learn a model, i.e. to estimate translation probabilities. Parallel corpora are different from normal text corpora in that they are not just a collection of texts, but are bi- or multilingual and structured so that every sentence is linked to its translation(s). Of course, researchers in SMT have neither the time nor the money to hire experts to translate tens or hundreds of thousands of sentences. This would also negate one of the major advantages of SMT, namely that a system can be adapted to new domains or languages pairs with relatively little effort. Instead, it is sensible to adapt pre-existing and openly available corpora to our purpose.

Early experiments on SMT were performed with the proceedings of the Canadian Parliament (Brown et al. 1990). Due to Canada's bilingualism, the proceedings of the Canadian Parliament are published in both French and English. These proceedings can be easily obtained and freely used. Additionally, the amount of text is extensive: Brown et al. (1990) report that they could obtain a corpus of 100 million words.

What is missing is a sentence-level alignment between the English and the French version of the proceedings. If we want to apply equation 1 to estimate translation probabilities from the corpus, we need to know which sentences are translations of each other, or *aligned*. It would be naive to assume that the *n*th sentence in the source text corresponds to the *n*th sentence in the target text. Brown, Lai and Mercer (1991) note that "at times a single sentence in one language is translated as two or more sentences in the other language. At other times a sentence, or even a whole passage, may be missing from one or the other of the corpora."

The proceedings of the Canadian Parliament are practical for sentence alignment in that they contain meta-textual information that can be used as anchor points. These include "session numbers, names of speakers, time stamps, question numbers, and indications of the original language in which each speech was delivered" (Brown, Lai and Mercer 1991). Since these comments are the same in both language versions of the pro-

ceedings, they can be used for a first alignment. If there are several sentences between two anchors, a sentence-level alignment is conducted based on the number of tokens the sentences contain. Brown, Lai and Mercer (1991) report that this method leads to an accuracy "in excess of 99%".

Apart from the availability of the different corpora and the feasibility of sentence-level alignment, there are various other selection criteria that need to be considered when choosing a corpus for SMT. There are obvious ones such as the quality and style of the translation. Ideally, a training corpus should be as similar as possible to the type of text we wish to translate. If a SMT system is used to translate texts on other topics than the training data, and are written in a different style, its performance will suffer. Experiments have shown that better results can be achieved with a smaller in-domain training corpus than a larger out-of-domain one (Koehn 2002). For research purposes, we are usually not bound to a specific text type. Consequently, we can circumvent this problem by selecting the training data first and then choosing a test set that fits the training data. Usually, part of the original corpus is held out from training so that it can be used for testing.

Today, the parallel corpus most frequently used in research is *Europarl*, which is based on the proceedings of the European Parliament (Koehn 2005). At the time the corpus was built, the proceedings were published in 11 languages, so that it can be used to train translation systems for 110 language pairs. The quality of sentence-level alignment is high thanks to clear and frequent anchor points. We will use Europarl to work on the language pair English – German. The corpus contains about one million sentences (35 million words) and will be described in more detail in the experiments section.

Now that we know the size of a typical training corpus, we can judge the effectiveness of a sentence-level MT system. Using approximately 1,000,000 sentences of Europarl as a training corpus, and 1000 random others as a test set, we found only 15 of the test sentences in the training corpus.[2] While we have so far only discussed how to translate a

[2]This number varies with the text genre: Hardmeier (2008) reports that sentences are often short and repetitive in the genre of film subtitles.

sentence that occurs in the training corpus, the real challenge lies in translating unseen sentences, that is, sentences that never occur in the training corpus. Considering the large proportion of unseen sentences in our test set, it is clear that these sentences cannot be simply ignored; nor is an increase in corpus size a promising and feasible solution. The problem of not having enough data is generally known as *data sparseness*, and is common to all statistical approaches. The solution is a more fine-grained system that calculates probabilities on a word level instead of a sentence level. An inspection of our test set shows that this makes sense. All but 50 of the 28,000 words in our test set also occur at least once in the training corpus. Of the 50 that do not, most are either proper names, which can be left untranslated, or hyphenated words.[3]

## 2.3 Word-Based Machine Translation

We have seen that sentences are ill-suited for probability estimations, and that a more fine-grained approach is sensible. Indeed, the first successful SMT systems worked on a word level (Brown et al. 1990). Interpreting a sentence as a sequence of words, we can rephrase the translation probability as in equation 2, $s_n$ and $t_n$ being the individual words in the source and target sentences $S$ and $T$:

$$P(T|S) = P(t_1, t_2, ..., t_n|s_1, s_2, ..., s_n) \tag{2}$$

The next central step is to apply Bayes' theorem:

$$\begin{aligned} P(T|S) &= \frac{P(S|T) * P(T)}{P(S)} \\ &= \frac{P(s_1, s_2, ..., s_n|t_1, t_2, ..., t_n) \cdot P(t_1, t_2, ..., t_n)}{P(s_1, s_2, ..., s_n)} \end{aligned} \tag{3}$$

[3]The portion of unseen surface forms is bigger when German is the source language, with 200 out of 26,000 test set words not occurring in the training corpus, including gems such as *Luftverkehrsknoten-punkten* (simply translated as *airports*) or *EU-Schulmilchprogramm* (*the EU's school milk programme*).

It may not be immediately apparent what benefit the Bayesian inference in equation 3 brings. The reason for applying it lies in the way the probabilities are approximated. Approximations of $P(T|S)$ are imperfect, though no more so than those of $P(S|T)$. $P(T)$ formalizes our expectation of a good sentence in the target language, and is easier to approximate than the translation probability in either direction. This will become clear when discussing how $P(S|T)$ is approximated in the *translation model* (*TM*), $P(T)$ in the *language model* (*LM*). Since we are interested in translating a given source sentence, which is the same for every potential translation, we can treat $P(S)$, the probability that someone utters $S$ in the source language, as a constant, and ignore it in our computation.

Equation 4 is universally true, but does not help us directly (see (Brown et al. 1990)). It illustrates the fact that the probability of a sentence can be reformulated as a chain of probabilities, the probability of each word depending on all preceding words.

$$P(t_1, t_2, ..., t_n) = P(t_1) \cdot P(t_2|t_1) \ldots \cdot P(t_n|t_1, t_2, \ldots, t_{n-1}) \tag{4}$$

The probability of a chain of events is easier to compute if the events are independent, i.e. if $P(t_n|t_1, t_2, \ldots, t_{n-1}) = P(t_n)$. This is true for an ideal die, for which the observation *1-2-3-4* is just as probable as *2-4-1-3*, but not for languages; intuitively, *all the world's a stage* is uttered more frequently, and hence more probable, than *stage world's the a all*. Still, we now make this independence assumption, also called Markov assumption, in order to counter data sparseness, however inaccurate it is for natural languages. For unseen sentences, we cannot estimate $P(t_n|t_1, t_2, \ldots, t_{n-1})$. $P(t_n)$, on the other hand, can easily be estimated through MLE, as long as $t_n$ is in the training corpus. In equation 5, we make the Markov assumption and rephrase the probability of a sentence as the product of the probabilities of all individual words, or unigrams.

$$P(t_1, t_2, ..., t_n) \approx \prod_{i=1}^{n} P(t_i) \tag{5}$$

A unigram-based language model ignores word order. If we assign the probability 0.1 to *a* and 0.01 to *stage*, *a stage* and *stage a* would return the same probability, namely 0.001. We expect this to be wrong, and we can improve the model by introducing $n$-grams of a larger order. Most language models are $n$-gram-based, an $n$-gram being a sequence of $n$ words. Conceptually, this means that the probability of a word $t_n$ in a sentence depends on the $(n-1)$ words preceding it. While longer $n$-grams are more informative, their drawback is that they are also more sparse. If no data for a given $n$-gram is found in the training corpus, lower order $n$-grams are used via back-off or interpolation (see (Stolcke 2002)). $N$-gram language models are efficient, robust and allow for the incorporation of larger monolingual corpora into the translation system (see (Federico and Cettolo 2007)).

More challenging than the language model is the *translation model* that estimates the probability $P(S|T)$. Generally, we can make the same independence assumption as in equation 5, approximating the translation probability as the product of $P(s_i|t_i)$. Firstly, however, we have to consider word alignment: it would be fatal to assume that the $n$th word in the source sentence is aligned with the $n$th word in the target sentence. Going back to examples 1 and 2, this naive assumption would lead to a translation model that considers German *die* a translation of *all*, and *ganze* a translation of *the*. Moreover, one word in the source language may be translated with several words in the target language, and vice versa. A typical example for this are German composite words such as *Bühnenanweisung*, which is translated as *stage direction*.

Brown et al. (1993) have proposed the first models for word alignment. The alignment algorithm takes into account the count of word pairs, sentence length and word order to estimate the alignment. Brown et al. (1993) stress that their algorithm "involves no explicit knowledge of either French or English". All alignments are discovered with a language-independent algorithm. While research on better word alignment models has continued in the last 15 years (Vogel, Ney and Tillmann 1996; Och and Ney 2003; Fraser and Marcu 2007), these so-called IBM Models introduced in the early

1990s are still influential today and are implemented in GIZA++ (Och and Ney 2003), which is considered state-of-the-art (Fraser and Marcu 2007).

The IBM models calculate word alignment in each translation direction independently. By intersecting the two alignments, we can increase precision at the cost of recall; by taking the union, we get a high-recall alignment with low precision. Different heuristics have been applied to make best use of the bidirectional alignment (see (Koehn, Och and Marcu 2003)).

Having obtained a word-level alignment between the two languages, we can now estimate the word translation probabilities analogous to equation 1. The only modification we have to make is that we do not count actual occurrences in the corpus, but alignment pairs. A word can have several alignments (or be unaligned), so that the number of aligned word pairs that include $t$ can be different from $count(t)$. Equation 6 takes this into account and expresses the probability of a word pair $(s, t)$ as the frequency of that word pair, divided by the frequency of all word pairs that include $t$, regardless of the source language word.

$$P(s|t) \approx \frac{count(s,t)}{\sum_{s'} count(s',t)} \tag{6}$$

Now, we need to consider differences in word order a second time. Given a perfect word alignment, translating example 1 word-by-word would result in this sentence: *Ganze die Welt ist eine Bühne*. What we need to do is to allow for an amount of *distortion* or *reordering* in our translations. We introduce a distortion parameter $P_D(S, T)$ to our model that penalizes the probability of a translation depending on the number of reordering steps made. This parameter is balanced by the language model, which hopefully prefers *die ganze Welt* over *ganze die Welt*. Without a language model, zero distortion is always preferred.

So far, the discussion of the different models has been focused on the training phase. In the so-called decoding phase, where we want to use existing models to actually translate a sentence, we can use related formulas. The main difference is that we no longer

want to find out $P(T|S)$ for two given sentences $T$ and $S$, but $\hat{T}$, that is the most probable sentence in the target language for a given source sentence $S$. This leads to equation 7.

$$\hat{T} = \arg\max_{T}(P_{LM}(T) * P_{TM}(S|T) * P_{D}(S,T)) \tag{7}$$

Modern SMT systems are built on this statistical foundation. We will now discuss the implications that word-based systems and the need of computing word alignment have for the use of parallel corpora, and then describe how this word-based approach was improved in state-of-the-art systems.

## 2.4 Parallel Corpora and Word Alignment

In the first discussion of parallel corpora, we saw that sentence alignment is a *conditio sine qua non* for using parallel corpora for SMT. However, sentence alignment is not sufficient when decomposing the translation of a sentence into the translation of its individual words. This leads to new problems.

First, let us consider the effect of non-literal translations. The Europarl corpus contains the following sentence pair:

4. This week has seen some 1,400 jobs lost in the footwear industry in England to low-cost producers overseas.

5. Just in dieser Woche gingen ca. 1400 Arbeitsplätze in der englischen Schuhindustrie an Billigproduzenten in Übersee verloren.

This translation is fine on a sentence level. If we look at the translation of the English phrase *has seen*, we notice that there is no literal translation in the German sentence. In such cases, there is a strong likelihood that word alignments that we intuitively consider wrong are learned in the training phase, such as $(has\ seen|gingen)$. This generally decreases the quality of a translation model.[4]

One of the disadvantages of Europarl is that the sentences are long, the average being approximately 26 tokens per sentence in the German corpus (after tokenisation). When working with the Europarl corpus, it is recommended to ignore the longest sentences in the training phase, since establishing a word alignment becomes more computationally expensive and less accurate for long sentences. By filtering out all sentences containing more than 40 tokens, the sentence average can be brought down to 20 tokens per sentence.

This is still a lot more than in other genres. Hardmeier (2008) reports that film subtitles are typically short, with an average of 11 tokens per subtitle, because they have to be readable in the short time that they appear on the television screen.

[4] For input sentences that are identical or very similar to the training material, an alignment we consider wrong can actually produce a good translation.

Not all problems are stylistic in nature. There are also linguistic phenomena that impede word alignment, for instance discontiguous phrases (see (Bod 2007)). In German V2 clauses (main verb in second position), verbal particles occur in clause-final position. In between the main verb and the verbal particle, any number of other words may occur. This is illustrated in example 6.

6. We *propose* a temporary Commission [...]

   Wir *schlagen* eine provisorische Kommission *vor* [...]

In theory, the IBM word alignment algorithm can align one word in English with a German word sequence of arbitrary length. However, it is not able to do so if the German phrase is discontiguous. Neither the alignment pair ($schlagen|propose$) nor ($schlagen\ eine\ provisorische\ Kommission\ vor|propose$) is satisfactory. One approach to deal with this problem is rule-based reordering, which we will discuss later.

## 2.5 State-of-the-Art in SMT: the Moses Toolkit

We use Moses for our experiments. Moses is an open source toolkit that includes all components necessary to build a phrase-based SMT system (Koehn et al. 2007), only relying on external tools for the creation of the language model and word alignment. There have been some important developments in SMT since (Brown et al. 1990) that have found their way into the Moses system. We will lay out some key changes in the following sections.

### 2.5.1 Log-Linear Models

We know that the methods for obtaining the different models in equation 7 are only approximations. Och and Ney (2001) therefore suggest that there is no reason to assume all system components should be weighted equally. New combinations of the different models can be achieved through log-linear modeling, which increases the flexibility of the system. Specifically, each model component $h$, for instance the language model, is

modified by a model parameter $\lambda$, which determines its weight on the result (Koehn and Hoang 2007).

$$\begin{aligned} \hat{T} &= \arg\max_{T} \prod_{i=1}^{n} h_i(T,S)^{\lambda_i} \\ &= \arg\max_{T} \sum_{i=1}^{n} \lambda_i * \log_e h_i(T,S) \end{aligned} \tag{8}$$

This allows for the inclusion of additional features that need not be mathematically justified, as long as they improve the results. For instance, the translation model $P_{TM}(T|S)$, which has been replaced with $P_{TM}(S|T)$ due to the application of Bayes' theorem, is reintroduced as an additional feature. Och and Ney (2001) list the inclusion of additional language models, of lexical resources, or a sentence length feature, as possible uses of this generalization.

An SMT system built with the default settings of Moses combines 14 different components. 5 components are provided by the translation model, these being the phrase translation probability in both directions, the lexical weight in both directions[5], and a constant phrase penalty that favours translations that use longer, but fewer phrases. The distortion model contributes 7 components, and the remaining two are the LM and a word penalty that controls the length of the translation.

Log-linear models are a great framework to experiment with new model components. Their only drawback is that they rely on a method to find the best parameters $\lambda$. How good the parameters are can only be determined if we have a way of measuring the quality of the resulting translations. Hence, this discussion will take place in the chapter on translation quality, chapter 3.

[5] See section 2.5.2 for details.

### 2.5.2 Phrase-Based Translation

Intuitively, we know that the translation of a word depends on its context. A word-for-word translation of *take the floor* does not convey the intended meaning. The phrase is not meant literally, but rather in the sense of *give a speech*. Translating the phrase word-by-word into German leads to *den Boden nehmen*, which does not convey the right meaning. A good translation into German is *das Wort ergreifen* (literally: *grab the word*). Although such idiomatic expressions are extreme examples, word-for-words translations are rarely ideal. Even the translation of common words such as *heavy* is highly ambiguous, its possible translations being *schwer*, *stark*, or *heftig*, among others. Consider figure 2.1 for examples in context.

| English phrase | German translation |
|---|---|
| a heavy stone | ein schwerer Stein |
| heavy traffic | starker/dichter Verkehr |
| heavy resistance | grosser/heftiger Widerstand |
| a heavy fine | eine hohe Bußе |
| heavy pressure | grosser Druck |

Figure 2.1: Different translations of *heavy* depending on context.

In word-based SMT, the translation model is built with the assumption that the individual word translations are independent of each other, which is clearly inadequate.[6] Koehn, Och and Marcu (2003) introduced a partial solution to this problem by modifying the translation model so that not only words, but also longer phrases can be treated as single units of translation. A phrase, in the context of phrase-based SMT, is not a linguistically motivated unit, but can be any sequence of words.

The phrase alignment is extracted from the word alignment data: "We collect all aligned phrase pairs that are consistent with the word alignment: The words in a legal phrase pair are only aligned to each other, and not to words outside" (Koehn, Och and

[6] Context is considered in the language model, and consequently, the right translation of *heavy* might be picked for the examples in figure 2.1. Still, trying to improve the translation model directly is a good idea.

Marcu 2003). This heuristic is simple to implement and has been shown to produce good results.

Formally, we no longer decode the source sentence S into $n$ words $s_1^n$, but into I phrases of arbitrary length $\bar{s}_1^I$. The phrase translation probability is expressed as $\phi(\bar{s}_i|\bar{t}_i)$, which leads to a new formula for the translation model:

$$P_{TM}(\bar{s}_1^I|\bar{t}_1^I) = \prod_{i=1}^{I} \phi(\bar{s}_i|\bar{t}_i) \tag{9}$$

The distortion parameter is similarly adjusted to work on a phrase level. The phrase translation probability is still estimated by MLE, as in equation 6.

The quality of phrase pairs is further validated by a lexical weight $lex(\bar{s}|\bar{t})$. The central idea of the lexical weight is to check if the individual words of a phrase pair are good translations of each other. The lexical weight of a word pair is estimated like $\phi(\bar{s}|\bar{t})$ by MLE (see equation 6).[7] The lexical weight of phrase pairs is then calculated as the product of the lexical weight of all aligned word pairs. As a simple example, the lexical weight for the phrase pair $(the\ dog|der\ hund)$ (*the* being aligned with *der* and *dog* with *hund*) equals $lex(the|der) \cdot lex(dog|hund)$. For source phrase words that are aligned with several target phrase words, we do not multiply, but take the arithmetic mean of the lexical weights (see (Koehn, Och and Marcu 2003) for more on the calculation of lexical weights). The lexical weight is added as an additional component to the log-linear model and improves BLEU score up to 1 point[8], compared to a phrase-based system without lexical weighting.

Phrase alignment is superior to word alignment, but inherits some of its limitations. Phrase alignment is blind to discontiguous phrases such as the one we have seen in

[7] For phrase pairs which consist of only one word on both the source and target side, we would expect $\phi(\bar{s}|\bar{t})$ to be equal to $lex(\bar{s}|\bar{t})$. This is not the case, however, because the two are based on slightly different alignments. $\phi(\bar{s}|\bar{t})$ is based on the intersection of the two mono-directional alignments produced by GIZA++, and ignores unaligned phrases. $lex(\bar{s}|\bar{t})$ uses the mono-directional alignment and includes unaligned phrases, and consequently has a larger denominator in equation 6. This means that $lex(\bar{s}|\bar{t})$ is equal or lower than $\phi(\bar{s}|\bar{t})$ if both phrases are of length 1.

[8] We report all scores on a scale from 0 to 100 (in contrast to a scale from 0 to 1, which is also common).

example 6, and will not translate them correctly in most instances.[9] Still, phrase-based SMT is the dominant paradigm in machine translation research at the time of this writing.

### 2.5.3 Factored Models

One application of log-linear models is the introduction of *factored translation models* (Koehn and Hoang 2007). Factored models are a framework that allow the integration of linguistic information into SMT. In this framework, words are represented as a vector of factors *factor0*|*factor1*|*factor2...*, for example *häuser*|*haus*|*NN*, factor0 being the surface form, the additional factors its lemma and POS tag. Technically, this vector is treated like a normal string in each model. The important characteristic of the factored framework is that we can define which factors to use in which step of the training or decoding process. For example, word alignment is usually performed on surface forms, while the translation model may include several factors on either (or both) language sides, or several translation steps with one factor each.

Along a translation path, one can freely combine several translation or language models that use different factors, since components that do not improve translation quality will receive a low weight during training.[10] One of the first proposed applications of factored models was the addition of a language model on the level of POS tags (Koehn and Hoang 2007), which requires a translation from surface word forms to target words consisting of two factors, the surface form and its POS tag.

Optionally, we can set the system to use alternative translation paths. Phrase translations will then be taken from either TM, and scored separately. Which translation path is preferred is determined by the weights. Koehn and Hoang (2007) implemented an analytical model that translates word lemma and morphology separately, then generates the target language surface form from these two factors. This may help if the word

[9] The exception being when long phrase pairs such as (*propose a temporary commission*|*schlagen eine provisorische kommission vor*) coincide with the source phrase that is to be decoded.

[10] It is important that a phrase pair occurs in each TM, otherwise it will not be used, even if the weight of a TM is 0.

*häuser* is unknown to the model, but *haus* is known. The analytical model has led to lower scores if used instead of the direct translation model, but an improvement could be achieved through alternative translation paths.[11]

### 2.5.4 Moses System Architecture

Having discussed the most important changes to the core of SMT, the algorithm that calculates translation probabilities, let us now elaborate on the more peripheral components of an SMT system such as Moses. Figure 2.2 shows the workflow during training (from the corpora to the different models) and during decoding (from input to output). It does not include optional steps/components such as syntactic parsing in preprocessing or additional target language corpora. The model parameters are typically computed through *Minimum Error Rate Training* (see section 3.2, page 43).

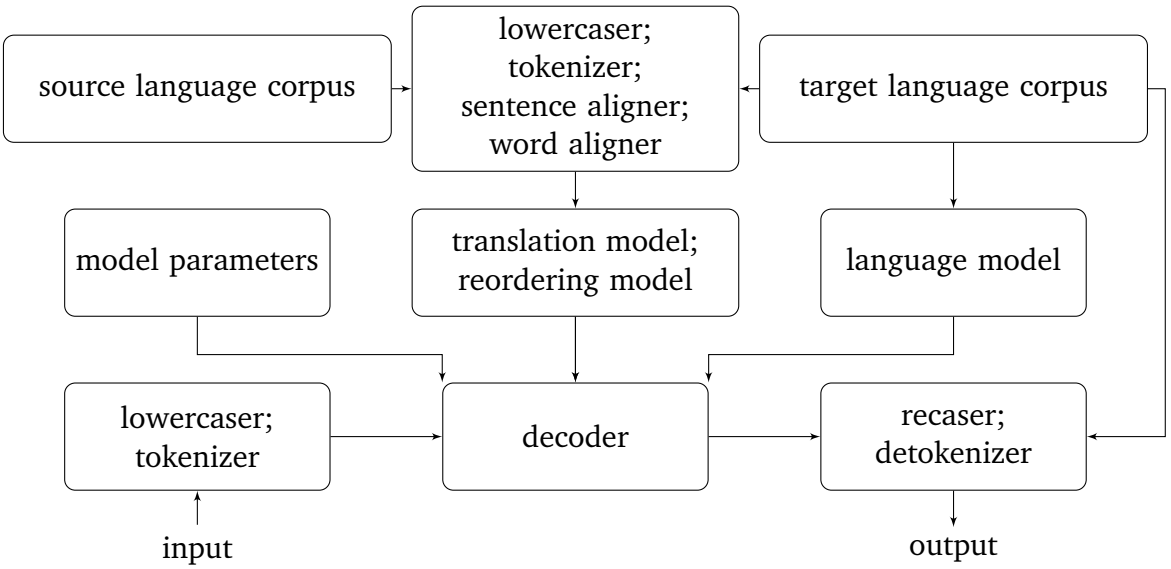


Figure 2.2: Moses flowchart

[11] While the weights were not discussed in Koehn and Hoang 2007, this would be achieved by setting high weights for the analytical translation model. Since high weights lead to low probabilities, this ensures that the analytical translation path is only used if the direct translation model yields no result.

We have already established how we use models to estimate the translation probability for a given sentence pair, but it would be inefficient to randomly generate possible output sentences and compute their score from the models. The Moses decoder employs a beam search algorithm (Koehn, Och and Marcu 2003). The algorithm generates output sentences from left to right, sorting the partial translations (hypotheses) according to their probability estimate. Any hypothesis that is not among the $n$ best is discarded to speed up decoding. While decoding is an integral part of SMT, a detailed discussion is beyond the scope of this thesis; the decoder is not affected by our experiments and does not contribute to changes in translation quality.

Before the input sentence is processed by the decoder, it is typically lowercased and tokenized. Both the lowercaser and the tokenizer are rule-based, and the tokenizer may include language-specific rules, typically lists of abbreviations that should not be tokenized. After decoding, a detokenizer and recaser are used to restore untokenized and mixed case text. While the detokenizer is again rule-based and mostly language independent, capitalisation is more difficult to predict. At the core of the recaser are a translation and language model trained on both the original and the lowercased target language corpus. Recasing is thus treated as a translation problem, with the difference that no reordering is necessary and that word and sentence alignment are trivial. Additionally to the statistical component, the recaser contains capitalisation rules such as always uppercasing the initial word in a sentence. These rules may be language-specific.

(De-)tokenizer, lowercaser and recaser are the only rule-based components of the Moses system, and the only components that are language-specific, even though the number of language-specific rules is low.

# 3 Translation Quality

## 3.1 Measuring Translation Quality

Any scientific theory has to be supported by hard facts. In a goal-oriented research field such as machine translation, we have to prove the validity of new approaches by showing that they improve translation quality, which is the main goal of MT research.[1] Consequently, research is driven by evaluation results: approaches that improve translation quality become more wide-spread, and others are abandoned. While the scientific discourse should not be limited to the discussion of results, it is a good thing that the evolution of machine translation systems is driven by merit rather than anything else.

Let us first define some basic criteria that any potential metric of translation quality has to meet. Estrella (2008), who provides an overview over different evaluation metrics, discusses several requirements: on a basic level, an evaluation metric has to be meaningful, that is, its results should correlate with the actual characteristic it measures. It should also be *objective* (independent of the opinions of the evaluators), *repeatable* and *reproducible* (a second evaluation of a system in the same environment should produce similar results), and *reliable* (free from random error).

### 3.1.1 Human vs. Automatic Evaluation

Originally, the task of measuring translation quality was exclusively performed by human judges. Designing an evaluation methodology that meets all the criteria laid out in the previous section has proven difficult. Simply asking untrained people for their

[1] Of course, maximizing translation quality is not the exclusive goal: efficiency and adaptability are also important characteristics of MT systems.

opinion on a translation will not lead to objective results. In order to successfully use human judgement in an evaluation, the characteristics that they have to evaluate, and the scale which they should use, have to be well-defined.

White et al. (1994) suggested three aspects of translation quality, *adequacy*, *fluency* and *informativeness*, that can be evaluated independently and, so they hoped, consistently. Evaluating a translation was thus decomposed into judging how well the translation represents the source text (adequacy), to what degree the translation is a well-formed, correct sentence (fluency), and if the information of the translation is comprehensible for readers (informativeness).

A common indicator for the reliability of human evaluations are inter- and intra-annotator agreement. Inter-annotator agreement is measured by making different annotators evaluate the same set of sentences, intra-annotator agreement by randomly repeating some sentences, so that they are evaluated twice by the same judge. In a recent evaluation, inter-annotator agreement in sentence ranking was found to be fair (around 0.37), while intra-annotator agreement was moderate (around 0.54) (Callison-Burch et al. 2008).[2] Callison-Burch et al. (2008) call the agreement "lower than [they] would like it to be" and suggest refining the instructions given to annotators in order to improve it.

Considering how difficult it is to make annotators judge sentences consistently, more indirect metrics have been proposed. These include comprehension-based metrics for which the human judges are asked "to read the machine translations and perform a task that shows how much they understood from the translations", and time-based metrics, where one measures "the time it takes a subject to execute a task with machine translated texts, such as reading or correcting the texts" (Estrella 2008).

Regardless of how reliable human evaluations are, they are all expensive, both in terms of time and money. Papineni et al. (2001) point out that "human evaluations can take months to finish and involve human labor that can not be reused". During

[2]The agreement is measured on a scale from 0 (no agreement) to 1 (perfect agreement), computed using the kappa coefficient ($\kappa$).

the development process, it is necessary to have quick and regular feedback on whether modifications of the system improve its quality or not. Human evaluations cannot meet this demand.

As a solution, Papineni et al. (2001) proposed BLEU, an automatic evaluation metric. The central idea behind BLEU and other automatic metrics is to measure the similarity of the MT output to one or several human reference translations. The underlying hypothesis is that "the closer a machine translation is to a professional human translation, the better it is" (Papineni et al. 2001). The score itself is a combination of $n$-gram counts performed on the system and reference translations.

We will now discuss the methodology of $n$-gram metrics in more detail, since they will be used for the evaluation of the experiments conducted for this thesis. A human evaluation, although still considered superior if performed correctly, is not an option for time and budget reasons.

### 3.1.2 $N$-Gram-Based Metrics: BLEU and NIST

BLEU (Papineni et al. 2001) is the most widely used method for automatic evaluation of machine translation. As previously mentioned, it is based on the comparison of the MT output with one or more human reference translations. Regardless of how this comparison is performed, relying on reference translations is error-prone. Two translations that are both good in terms of fluency and adequacy can still be starkly different on a surface level. Consider the following two translations of the source sentence (example 7). Example 8 is the human reference translation; example 9 is a MT system output[3]:

7. We know all too well that the present Treaties are inadequate [...]

8. Uns ist sehr wohl bewusst, daß die geltenden Verträge unzulänglich sind [...]

9. Wir wissen nur zu gut, dass die gegenwärtigen Verträge nicht ausreichen [...]

[3] taken from `http://www.statmt.org/matrix/`

While both translations are perfectly adequate, they have little in common on the word level, sharing only two of eleven tokens. Examples 8 and 9 illustrate that there are many valid ways to express the same meaning. *N*-gram-based metrics are sensitive to even small differences between system and reference translation. If a token in the system translation is not found in the reference translation, the token is considered wrong, even if the only difference is a spelling variation (*dass* vs. *daß*) or the ending of an otherwise correct word.[4]

We are able to use several reference translations to increase the probability that a good system output corresponds to at least one reference translation. Also, the efficiency of BLEU allows for an increase in test corpus size, so that random effects are averaged out. An increase in test size is not only possible, but also necessary because of BLEU's poor performance on single sentences. Only when comparing two systems based on their respective translation of a large test set does the score correlate with human judgement.

On a technical level, BLEU counts the *n*-gram precision of the translation that is evaluated (system translation). For every unigram in the system translation, the algorithm checks whether this unigram also occurs in one of the reference translations. If a unigram occurs multiple times in the system translation, it is only counted as often as it occurs in the reference translation (clipping). The same is then repeated with *n*-grams of any length (usually up to four), and their geometric mean is taken as the result. Unigram precision disregards word order, while longer *n*-grams penalize translations that contain the right words, but in the wrong order. Hence, Papineni et al. (2001) suggest that unigram scores are a better indicator of adequacy, while longer *n*-grams correlate with fluency.

One shortcoming of BLEU's precision-based approach is that a shorter translation may have high precision scores, even if they leave part of the source sentence untranslated.

[4] We suspect that this makes BLEU biased against morphologically rich languages. It is unclear how much this bias contributes to the poor results of translations into Finnish and German in (Koehn 2005), but the fact that Finnish also underperforms as source language shows that morphologically rich languages are indeed difficult to translate, not only to evaluate.

In BLEU, this is compensated with the so-called brevity penalty. The low recall of such a translation is not directly measured in the BLEU score, since measuring recall is only possible for individual reference translations, not entire sets of reference translations. Instead, the brevity penalty makes sure that translations that are shorter than the shortest reference translation receive a lower score.

While BLEU does correlate well with human judgement (Papineni et al. 2001; Banerjee and Lavie 2005), alternative metrics were proposed that eliminated some of BLEU's weaknesses. The NIST score introduced by Doddington (2002) is a derivative of the BLEU scoring algorithm which adds a weighting of the $n$-grams according to their frequency, weighting rare $n$-grams more heavily for the score. Additionally, the brevity penalty has been modified and the arithmetic means of the $n$-gram scores is taken instead of the geometric mean.

### 3.1.3 $N$-Gram-Based Metrics: METEOR

One criticism that is left unaddressed by these modifications to BLEU is the fact that the BLEU/NIST metrics do not consider recall at all. Banerjee and Lavie (2005) point out that recall is an important indicator of "to what degree the translation covers the entire content of the translated sentence". They propose the METEOR metric that includes a calculation of recall. The problem of several reference translations being available, which caused Papineni et al. (2001) to ignore recall, is solved by scoring recall for each reference independently, and using the best score. METEOR is unigram-based and does not use higher-order $n$-grams to measure fluency. Instead, Banerjee and Lavie (2005) introduce a new measure of fragmentation that counts the number of chunks, i.e. word sequences, that occur in both the reference translation and the system translation. Fewer and longer chunks gain a higher score than many short ones. Banerjee and Lavie (2005) report that METEOR correlates significantly better than both BLEU and NIST with human judgement on a system level.

### 3.1.4 Caveats for the Use of Automatic Metrics

There are voices that criticize the over-reliance of the MT community on BLEU and similar automatic scores. A case in point is that a rule-based MT system obtained vastly lower BLEU scores than SMT systems, even though the rule-based system was preferred in a human evaluation (Callison-Burch, Osborne and Koehn 2006). This poses the danger that research is misdirected towards approaches that maximize BLEU scores instead of translation quality. We can provide our own example of such an approach. We find various spelling variations in the German section of Europarl, most notably *dass* and *daß*. By normalizing the two forms to the one used in the test set, we were able to increase the BLEU score by 0.7 points. Yet this normalization has no effect on how a human would evaluate the translation.

Callison and Burch (2006) conclude that automatic evaluation metrics, while valuable because of their being fast and inexpensive, should not be over-used:

> Appropriate uses for Bleu include tracking broad, incremental changes to a single system, comparing systems which employ similar translation strategies (such as comparing phrase-based statistical machine translation systems with other phrase-based statistical machine translation systems), and using Bleu as an objective function to optimize the values of parameters such as feature weights in log linear translation models, until a better metric has been proposed.
>
> Inappropriate uses for Bleu include comparing systems which employ radically different strategies (especially comparing phrase-based statistical machine translation systems against systems that do not employ similar *n*-gram-based approaches), trying to detect improvements for aspects of translation that are not modeled well by Bleu, and monitoring improvements that occur infrequently within a test corpus. (Callison-Burch, Osborne and Koehn 2006)

There is a risk that the experiments conducted for this thesis fall under the category of inappropriate uses, since some of the phenomena that benefit from syntactic information are too infrequent to have an impact on BLEU scores. We will thus not only perform a quantitative evaluation of the systems, but also provide an in-depth analysis of selected translations to see if the experimental changes have the desired effect.

## 3.2 Optimizing Weights: Minimum Error Rate Training

In the introduction to phrase-based SMT, we saw that the different system components are modified by weights so that their relative contribution to the system can be optimized. A procedure to determine the optimal weights, called *Minimum Error Rate Training* (MERT), was proposed shortly after the development of automatic evaluation metrics (Och 2003).

The idea behind MERT is simple: we let a system translate a set of sentences (*development set*) several times with different weights, score each output with an evaluation metric (typically BLEU), and then pick the weights that result in the best scores. However, finding optimal weights, and doing so efficiently, is a great challenge. We can illustrate MERT as the search for the best coordinates in an $n$-dimensional weight space, $n$ being the number of weights that have to be set.[5] Assuming that we choose a grid of $m$ weights per dimension, we would have to translate our development set $m^n$ times to test all possible weight combinations.

Since translating is the most time-consuming part of MERT, current implementations of MERT approximate the decoder output through generating $n$-best lists, i.e. lists of the $n$ best translations candidates with a given weight. Additionally, instead of testing all weights (or a random selection), each new iteration uses the weights that are most likely to lead to better scores considering prior iterations.

[5] A vanilla Moses system has 14 components. Alternative translation paths and additional models further increase the number of weights needed.

Bertoldi, Haddow and Fouet (2009) point out the limitations of this approach: "Because the error surface is highly nonconvex, MERT is always at risk of being trapped at local maxima; and because it uses n-best lists as an approximation for the decoder output, it cannot explore the actual parameter space". An illustration of the non-convexity of the error surface of BLEU is shown in figure 3.1. Research to improve MERT is ongoing (Cer, Jurafsky and Manning 2008; Macherey et al. 2008; Bertoldi, Haddow and Fouet 2009; Foster and Kuhn 2009).

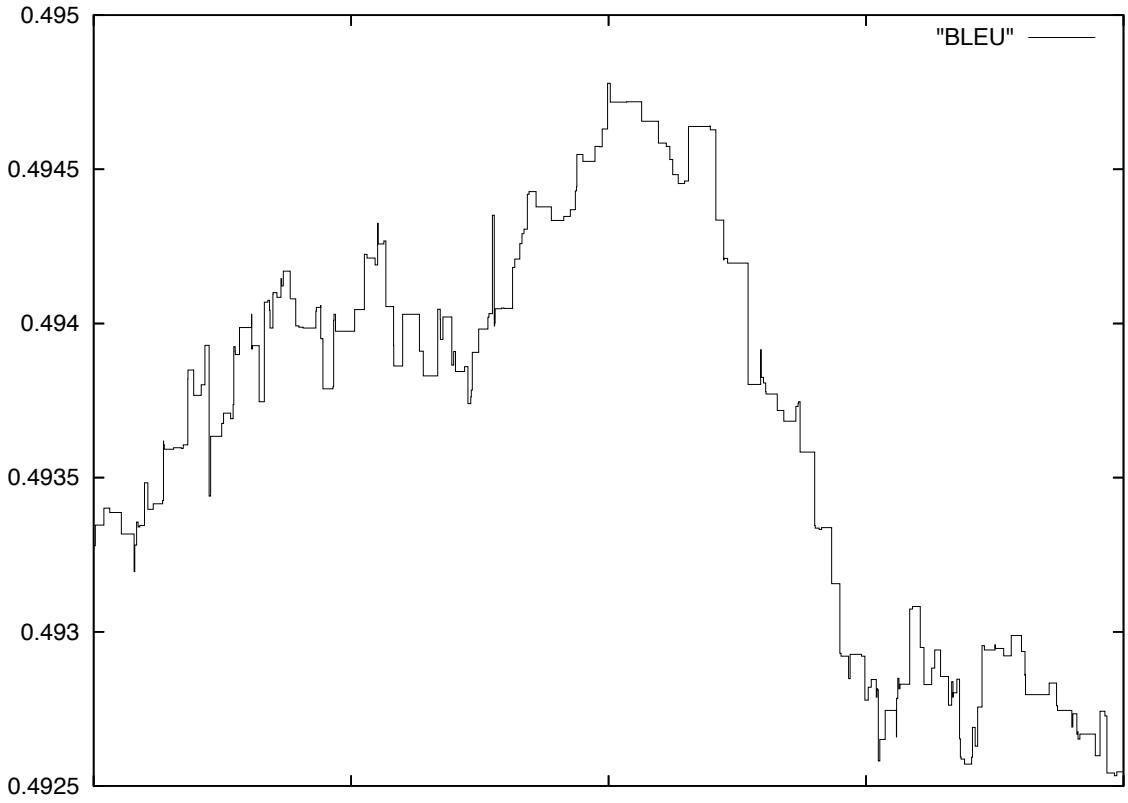


Figure 3.1: BLEU error surface when varying one parameter. Figure from (Arun 2007).

Despite its limitations, MERT leads to a tremendous improvement in BLEU score. Och (2003) reports an increase in test set BLEU score from 11.3% to 17.2% in the translation of news text from Chinese to English. The baseline was established through *Maximum Mutual Information* training, the standard method of training weights before the introduction of MERT (Och and Ney 2001). This does not mean that the impact on actual

translation quality is equally great: “It might happen that by directly optimizing an error measure in the way described above, weaknesses in the measure might be exploited that could yield better scores without improved translation quality” (Och 2003).

We have discussed MERT more extensively than other system components that are more crucial for SMT. The reason for this is that sentence/word alignment and the language model remain unchanged for all or most experiments described in this thesis, and thus have no direct effect on differences in score between two experimental systems. In contrast, every experimental system needs its own set of weights; using the baseline weights for experimental systems is not possible because of the different number of weights; if it were, it would yield suboptimal results. The fact that MERT can only approximate the best weights brings a level of uncertainty to all evaluation results. When running MERT ten times, the difference between the best and the worst result can be one BLEU point or more (Arun 2007; Foster and Kuhn 2009). A second problem is that “two weight vectors that give approximately the same dev-set BLEU score can give very different test-set scores”, according to Foster and Kuhn (2009).

When finding significant differences in score, we cannot determine without doubt whether they are caused by MERT or by true improvement/deterioration of the underlying system, unless the difference is large – larger than improvements usually reported in this field. In order to limit variation, Foster and Kuhn (2009) suggest running MERT at least seven times and reporting the best scores. This is time-consuming and was not possible for this thesis, but we have run MERT at least twice for any system evaluated, reporting the better scores.

# 4 Using Syntax in Machine Translation

## 4.1 Linguistically Motivated Modifications of Statistical Machine Translation: General Remarks and Related Work

While the focus of this thesis is on the usage of syntactic information to improve a phrase-based SMT system, we would like to contrast this approach with others that are linguistically motivated. A lot of research is focused on remedying current weaknesses of SMT systems with linguistic knowledge. In principle, this is possible with every component of an SMT system: word alignment, the translation model(s), the language model(s), word reordering etc.

Obtaining word alignment and phrase alignment are critical steps for SMT, which currently are usually performed without linguistic resources. Syntax-based alignment has been investigated, but phrase-based approaches have consistently outperformed purely syntax-based ones (Koehn, Och and Marcu 2003; Yamada and Knight 2001). Koehn et al. (2003) have found that limiting phrase translations to syntactic phrases is harmful to translation performance, because it eliminates valuable phrase pairs from the translation model. They illustrate this with the often-quoted phrase pair (*es gibt*|*there is*). While neither phrase is considered a syntactic phrase in constituency grammars, translating them as a unit yields better results.

Syntactic phrase pairs have been shown to be useful if they are incorporated into the translation model differently. By supplementing a phrase-based translation model with constituent pairs, rather than restricting it to constituents, Tinsley et al. (2007) report a significant improvement in BLEU score.

Tiedemann (2005) proposes a word alignment algorithm that includes GIZA++ alignment, but additionally takes into account linguistic *clues*. These clues can come from bilingual dictionaries or include "manually defined relations between similar linguistic features such as part-of-speech" (Tiedemann 2005). He reports an improvement in alignment of 6% points in F-value over the GIZA++ baseline on a manually constructed gold standard.

One of the problems common to all systems is data sparseness. Both word-based and phrase-based SMT systems are unable to translate unseen surface forms, and in morphologically rich languages such as German, these are more common than in English. With morphological analysis, the amount of unseen data can be reduced: splitting long German compounds into several separate words increases the likelihood that these occur in the training corpus (Koehn and Knight 2003). In a more recent experiment, a SMT system was built that translates lemmas and their morphological information independently (Koehn and Hoang 2007). The general direction of these approaches is the same as in the ubiquitous lowercasing of texts before they are processed by an SMT system: abstract from the surface form level to a level of representation with *types*, which makes each type more frequent. On the one hand, this reduces data sparseness. On the other, information is thus thrown away which may not be perfectly generated again. The system operating on a lemma level by Koehn and Hoang (2007) led to a significant drop in performance when it was used instead of the surface form model. Only by combining both models on alternative translation paths was an improvement achieved. Generally, these approaches have a larger positive effect if the training corpus is small.[1]

A diametrically opposite idea is not to reduce the number of types in a corpus, but to increase it. This is achieved by enriching the surface words with linguistic information, and representing this information as additional factors. Syntactic enrichment can help to distinguish ambiguous word forms, for instance by indicating whether a noun phrase

[1] Koehn and Hoang (2007) report an improvement with a German–English corpus consisting of 52,000 sentences, Hardmeier (2008) with 100,000 tokens, translating from Swedish to Danish. Using a corpus of 10,000,000 tokens, morphological analysis led to no significant improvement (Hardmeier 2008).

| Level of Representation | Tokens | Types |
|---|---|---|
| Surface form (tokenized and lowercased) | 18,505,811 | 215,528 |
| Lemma | 18,505,811 | 162,034 |
| Syntactic relation label | 18,505,811 | 41 |
| Surface form + syntactic relation | 18,505,811 | 449,046 |
| Surface form + syntactic head | 18,505,811 | 2,742,985 |

Table 4.1: Number of types and tokens in the German training corpus, depending on level of linguistic abstraction.

is subject or object, a distinction which is relevant for its translation into German. However, the combination of surface forms and syntactic factors increases the number of types in our corpus, which results in sparser data. This necessitates the combination of the specific model with a general one; otherwise, the loss in performance caused by data sparseness far overweighs any potential improvements. Table 4.1 shows the effect different levels of linguistic abstraction have on the number of types and tokens in the German training corpus. While the number of tokens remains constant, the number of types varies greatly.[2] It is easy to see that the number of types is far lower for the more general levels of linguistic abstraction. We can expect two benefits from this: on the one hand, we reduce the likelihood of unseen words; on the other, the average number of observations of each token is increased, which is beneficial for word alignment and MLE. In contrast, levels of representation that combine the surface level with other linguistic information are disadvantaged due to their sparsity. The value of the information added as factors has to outweigh this disadvantage in order to justify factorisation. Avramidis and Koehn (2008) show that noun case information gained from syntactic analysis helps translating from morphological poor languages such as English. Birch, Osborne and Koehn (2007) investigate the use of CCG supertags on the source side for German–English translations, although with inconclusive results.

[2]It is another interesting fact that there are only 71,810 types in our English training corpus, in contrast to the 215,528 in the German one. Composite words and the inflectional richness of German account for this fact.

While the approaches described so far mainly deal with translation models, one can also aim to improve language models. In part-of-speech tagging, every word is assigned one of a few dozen classes.[3] Due to the low number of types, the POS level of representation does not suffer from data sparseness to the same degree as the word form level. Thus, the probability of a sequence of words can be estimated even if it does not occur in the training corpus. Koehn and Hoang (2007) performed several experiments with additional language models. The improvement for the language pair English–German was small (0.16 BLEU percentage points). For the language pair English–Czech, however, an improvement of almost 2 BLEU percentage points was achieved.

Linguistic information is also used for preprocessing of the training corpora and translation input, which are then processed by an unmodified phrase-based SMT system. Usual preprocessing steps include lowercasing and tokenization. In order to restore the normal text form, recasing and detokenization is required in postprocessing. Preprocessing approaches do not necessarily have to aim at reducing the number of types. One wide-spread approach is word reordering, which aims to make the source language more similar to the target language in terms of word order ((Nießen and Ney 2004; Collins, Koehn and Kučerová 2005; Popović and Ney 2006; Holmqvist et al. 2009), to name a few). This benefits GIZA++ word alignment and consequently translation quality.

A reordering based on syntactic information allows more complex reordering rules, such as always placing the subject immediately before the verb complex in German (Collins, Koehn and Kučerová 2005), but POS-based reordering approaches have also shown to be successful (Popović and Ney 2006). Holmqvist et al. (2009) have also experimented with reordering the target language instead of the source language, but found that this resulted in lower BLEU scores.

Linguistic analysis has been used for reranking of the $n$-best lists that SMT systems generate. Hasan, Bender and Ney (2006) use shallow parsers to identify ungrammatical

[3] 54 in the German STTS tagset (Schiller et al. 1999).

hypotheses and rerank the the *n*-best list accordingly, resulting in an up to 0.7% points gain in BLEU score.

This thesis will focus on syntactic enrichment of the source and target language using factored models; the experiments are thus most similar to (Birch, Osborne and Koehn 2007; Koehn and Hoang 2007; Avramidis and Koehn 2008). Also, syntax-based target language reordering is investigated, bearing resemblance to (Collins, Koehn and Kučerová 2005; Holmqvist et al. 2009). The next sections will elaborate how syntactic enrichment can be of use for SMT.

## 4.2 What Can Syntactic Parsing Tell Us?

There is a plethora of competing syntactic theories, and a full discussion thereof would go beyond the scope of this thesis. In lieu, we will talk about the information we can gain from syntactic parsing in general. There are two main types of syntactic structure: the grouping of words and the relation between them.

*Constituency parsing* analyses the segmentation of a sentence into groups of words, or *phrases*, which, unlike the use of the term in phrase-based SMT, are not arbitrary word sequences, but must be linguistically sound units. Informally, we can test whether words form a group with simple tests. For instance, a noun phrase can be replaced by a single pronoun in a sentence to construct another grammatical sentence. Consider example 10:

10. The man sees the dog.

Since the sentence *the man sees him* is also grammatical, we can conclude that *the dog* forms a noun phrase. The same is not possible with **the man him* or **the man sees the him*. A constituency analysis takes the whole sentence as a starting point, segmenting it into different phrases, which in turn can be segmented into smaller phrases or terminal nodes, the actual words.[4]

[4]The whole sentence is a starting point in the representation of constituency structure. The parsing algorithm itself does not necessarily start from the top node.

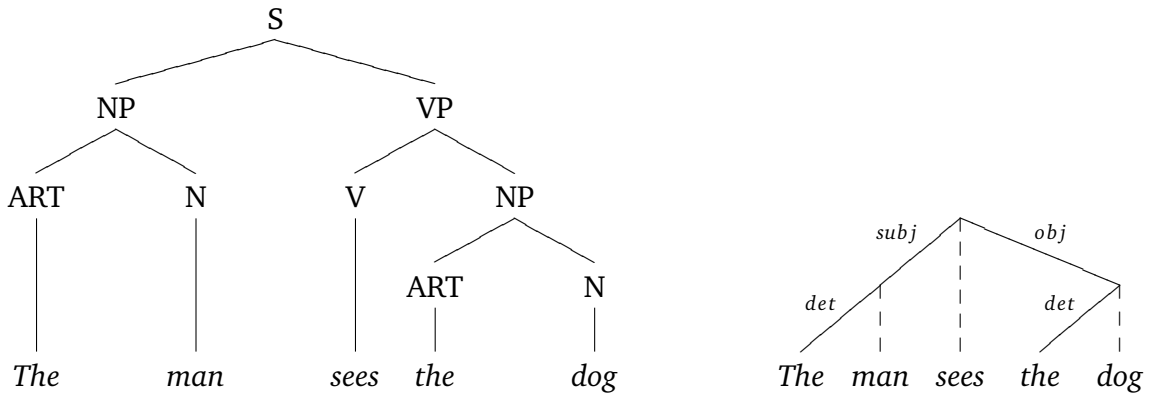


Figure 4.1: Prototypical parse trees in constituency (left) and dependency (right) structure.

*Dependency parsing* explores the relations between the words in a sentence. It takes its name from the notion that there is a hierarchy within each sentence, and that all words (except the *root*) depend on others. That such dependencies exist can be observed through agreement: in inflectionally rich languages such as German, articles and adjectives have to agree in case, number and gender with the noun they modify; they can also be said to depend on that noun. We call the dominant word in a relation the *head*. The root of a sentence is typically the finite verb. Figure 4.1 shows an analysis of example 10 in both constituency and dependency representation.

Constituency and dependency parsing have different potential uses for SMT. Since constituency parsing segments a sentence into smaller units, and we also need to segment a sentence into units for SMT, it seems obvious to use syntactic phrases as the main translation unit. In the last section, we have seen that this has proven less successful than phrase-based SMT, phrase-based in the sense that all word sequences compatible with word alignment are accepted as units. Dependency parsing is better compatible with factored translation models, since each word in a sentence is assigned one head and one relation type, which can be used directly as factors.

In practice, the two grammar approaches are quite reconcilable. Many grammars include both constituency and functional information, for instance the Lexical Functional

Grammar (Dalrymple 1999). Also, a mapping between constituency and dependency trees has shown to be possible as long as both grammars are similarly detailed (Johansson and Nugues 2007).

The level of detail in a syntactic analysis is a far-reaching design decision. Shallow parsing is typically faster and has to deal with fewer ambiguities than full or deep parsing, but may leave important distinctions underspecified. *Chunking* is a shallow parsing process that only identifies phrases, but not the relations between them. Among the information that is missing are syntactic roles. Noun phrases can serve as the subject or object of a sentence, among others. This has an effect on their translations, since the syntactic role is encoded differently in different languages. In German, the role is marked by case; in English, by word order.

A full syntactic analysis is more detailed, but this increase in detail is paid for by higher computational cost and, since more ambiguities have to be resolved, more classification errors. The possible advantage of a full syntactic analysis is clear. Every syntactic distinction might be relevant for a translation. However, it is also possible that some of them are not. And since an increase in the number of syntactic categories leads to more data sparseness, uninformative syntactic relations are undesirable. Considering parser choice, this still speaks in favour of full parsers. Even if we are only interested in partial analyses, these can be easily extracted from full ones.

We have chosen a full dependency parser, Pro3Gres, for our experiments (Schneider 2008). It is unlikely that any parser provides an output that is optimal for the specific SMT task. We will first show what information Pro3Gres provides. In the experimental section, we will then go into more detail as to what information is used, and how.

## 4.3 Using Pro3Gres in Statistical Machine Translation

We will now have a quick look at the Pro3Gres output and see how it affects the SMT system. Of the four experiments we will conduct, two require syntactic information

from the source text, two from the target text. We have used Pro3Gres, a full dependency parser with grammars for both English and German (Schneider 2008; Sennrich et al. 2009).

Pro3Gres is a robust and fast bi-lexicalised dependency parser originally developed for English. It uses a hybrid architecture combining a manually written functional dependency grammar (FDG) with statistical lexical disambiguation obtained from the Penn Treebank. Pro3Gres uses a context-free CYK parsing algorithm, but also models the majority of English long-distance dependencies. The parser delivers good performance on newspaper texts. Schneider (2008) reports about 84% F-value for subjects and direct objects on a newspaper test corpus. The German version of the parser uses data extracted from the TüBa-D/Z treebank for its statistical disambiguation (Telljohann, Hinrichs and Kübler 2004). It is competitive with state-of-the-art statistical parsers, and outperforms them in the prediction of central grammatical relations such as subjects and objects, thanks to the integration of morphological analysis. On a newspaper test corpus, it reaches an F-value of 82% for subjects and 74% for direct objects (Sennrich et al. 2009). We expect a slight decrease in performance on Europarl due to the fact that neither the German nor the English version of Pro3Gres is optimized for parsing of spoken texts.[5] Apart from its solid performance, an important reason for choosing Pro3Gres was our acquaintance with the parser, which allowed us to modify the parsing process to our needs. Specifically, we had to override tokenization, sentence boundary detection, and the analysis of secondary edges to optimally integrate it into our SMT system.

Regardless of the information we wish to add, it is important that no information is lost during parsing, or in other words, that we are able to reconstruct the original corpus from the parser output. Hardmeier (2008) highlights that this cannot be taken for granted. For example, a parser's sentence boundary detection may distort the sentence alignment, which is absolutely necessary for SMT. We will now describe the workflow of the Pro3Gres parser, and how we ensure that no information is lost.

[5] However, the proceedings are very formal and do not record speech phenomena that make parsing notoriously difficult, i.e. repetitions, false starts, fillers or cut-off utterances.

Pro3Gres requires preprocessing performed by several external tools, each of which has its own limitations (e.g. meta-characters) that may inadvertently distort the original text. The German Pro3Gres system relies on the TreeTagger for POS tagging, and Gertwol for a morphological analysis (Schmid 1994; Haapalainen and Majorin 1995). Normally, Pro3Gres relies on TreeTagger for tokenization and sentence boundary detection, but we disabled this behaviour. Consequently, tokenization and sentence boundaries were adopted from the unparsed text. The only difference between the unparsed text and the parsed one is that non-latin1 characters, such as €, were removed from the parsed text because the version of TreeTagger that was used does not support UTF-8 encoding.

```
sent([[man, 'NN', ['The_DT', man_NN], man], [see, 'VBZ',
[sees_VBZ], sees], [dog, 'NN', [the_DT, dog_NN], dog]]).

analyses(1, 17.1991, [], 1-3, ['see#2'('man#1'(['The_DT',
man_NN]), '<-subj<-', [sees_VBZ], '->obj->',
'dog#3'([the_DT, dog_NN]))]).

detmod1(_,'man#1',the).
detmod1(_,'dog#3',the).

subj('see#2', 'man#1', _, '(<-)').
obj('see#2', 'dog#3', _, '(->)').
```

Figure 4.2: Sample output of Pro3Gres for example 10.

The English Pro3Gres system requires tokenized, chunked and lemmatised input. All of these functions are performed by the LT-TTT2 toolkit, which includes the C&C tagger and the morpha lemmatiser (Minnen, Carroll and Pearce 2000; Curran and Clark 2003; Grover and Tobin 2006). Since tokenization is tightly integrated in the system, it could not easily be disabled. This leads to some differences in tokenization between the original text, which is tokenized with a script provided by the Moses system, and the parsed text. For example, *cannot* is split into two words by LT-TTT2, *can*

and *not*. However, these changes only affect the source text, and have little effect, if any, on the translation quality of the SMT system, as long as tokenization is consistent both in the training and the test set. Tokenization differences in the target text would be of greater consequence, since these could falsify the evaluation results. Apart from this difference, the original text can easily be reconstructed.

Figure 4.2 shows a sample output of the Pro3Gres system. Additionally to the surface forms, we have access to the head lemma of each chunk (e.g. *see*), POS tags and dependency relations. We will elaborate how this information is incorporated into the SMT system in the discussion of the individual experiments.

# 5 Experiments

## 5.1 Method

All experiments were carried out on the English–German language pair of the Europarl corpus (Koehn 2005). We split the corpus into three parts: the largest segment of about one million sentences served for training; the remaining 10% of the corpus were held out for tuning and testing. Out of these, a set of 1000 sentences was used for tuning the weights, and another 1000 sentences for the evaluation. The training sets for the language model and the translation model differ in that all unaligned sentences, and all sentences longer than 40 tokens, were removed for the training of the translation model. This is because word alignment becomes more computationally expensive and less reliable for long sentences. The exact number of tokens in each segment is given in

| subcorpus | sentences | tokens (English) | tokens (German) |
|---|---|---|---|
| training set (translation model) | 911,122 | 19,570,126 | 18,505,811 |
| training set (language model) | 1,147,490 | 31,899,096 | 30,252,375 |
| development set (MERT) | 1000 | 30,223 | 27,084 |
| test set (evaluation) | 1000 | 28,315 | 26,517 |

Table 5.1: Size of training, tuning and evaluation set for all experiments, including the baseline.

table 5.1.

The baseline system was built following the guideline for the EACL 2009 Workshop on Statistical Machine Translation [1], the only difference being that we used the training and test sets described above. We will now lay out in detail what tools we used for

[1] `http://www.statmt.org/wmt09/baseline.html`

the experiments. The main component of the SMT system is Moses, a state-of-the-art, phrase-based and factored SMT toolkit (Koehn et al. 2007). Moses contains most of the components needed to build an SMT system, including tools for preprocessing data, training models, MERT, and evaluating the results. Word alignment was calculated with the GIZA++ toolkit (Och and Ney 2003). We used only surface word forms to calculate word alignment, even when enriching the corpus with additional data. The language models were built with SRILM (Stolcke 2002). We obtained BLEU and BLEU-unigram precision scores with version 12 of the NIST scoring tool (Doddington 2002).[2] We used version 0.7 of the METEOR system to obtain the METEOR scores (Banerjee and Lavie 2005). Statistical significance of BLEU results was measured by bootstrap resampling (Koehn 2004). One needs to bear in mind, however, that the variation caused by MERT may surpass significance levels, which means that even statistically significant differences are not necessarily caused by a better or worse model.

We used the Pro3Gres parser for the syntactic analysis, and the preprocessing tools described in the last chapter. Since there are small differences between the unparsed and the parsed corpus, even on a surface form level, the parsed corpus was also used for the training of the baseline system. Thus, preprocessing can be ruled out as a cause of any changes in translation quality.

## 5.2 Baseline System

The last section already covers how the baseline system was built. In this section, we will use the baseline system to illustrate some important points regarding the interpretation of evaluation results.

Table 5.2 shows the results from three different baseline systems. The systems are identical except for the parameters set by MERT. Most importantly, we can observe

[2] BLEU scores vary depending on the script used and its version. Minor differences in scoring, for example how tokenization and casing are handled, may have a large impact on the results. Needless to say, these factors were kept constant for all experiments.

| Experiment | development set | | | test set | | |
|---|---|---|---|---|---|---|
| | BLEU | Unigrams | METEOR | BLEU | Unigrams | METEOR |
| Baseline system 1 | **18.91** | **51.64** | 23.63 | 16.63 | 49.14 | 21.47 |
| Baseline system 2 | 18.82 | 51.62 | **23.80** | **16.96** | **49.39** | 21.68 |
| Baseline system 3 | 18.75 | 51.51 | 23.71 | 16.75 | 49.29 | **21.83** |

Table 5.2: Baseline results with different model parameters (obtained by MERT). Best results in bold.

a variation of approximately 0.3 BLEU points[3] in the test set between the best and the worst system. Even though the difference is statistically significant ($p < 0.05$), the variation is caused entirely by MERT. Consequently, we cannot simply assume that statistical significance is sufficient evidence that an experimental system is inherently better (or worse) than the baseline.

Interestingly, the system that performed best on the development set obtained the worst test set scores. For this reason, both development and test set scores will be reported in the following sections. While some variation in the difference between development and test set scores can be attributed to chance, some systems might be more prone to overfitting. We know that MERT, as any machine learning technique, overfits the parameters, meaning that it finds parameters which maximize the score on seen data (the development set), but not on unseen data (the test set). Overfitting problems become more serious when optimizing a higher number of parameters (Och 2003). Since most experiments described in this thesis feature more parameters than the baseline system, we have to observe if overfitting becomes worse.

[3]We report all scores on a scale from 0 to 100 (in contrast to a scale from 0 to 1, which is also common).

## 5.3 Additional Input Factors

### 5.3.1 Syntactic Relation as Input Factor

#### Motivation

In English, a strict SVO (subject-verb-object) language, the syntactic roles are marked by word order. The noun phrase directly before the verb is typically subject, the one after the verb an object. While German is also considered an SVO language, it has a looser word order. Instead of word order, grammatical case is the primary marker for syntactic roles. This means that the same noun phrase in English has different translation equivalents in German, depending on its syntactic role. Let us consider the English phrase 'the man' in examples 11 and 12:

11. The man sees the dog.

    Der Mann sieht den Hund.

12. The dog sees the man.

    Der Hund sieht den Mann.

If *the man* is the subject of the verb *see*, its German translation is *der Mann*, the subject role being marked through the nominative case. In object position, *the man* is translated as *den Mann*, a noun phrase in the accusative case.

Phrase-based translation approaches rely on local context to determine the correct translation of an ambiguous phrase such as *the man*. This contextual information is restricted to the LM and long phrases in the TM. Our expectation is that *the man* in example 12 will only be translated correctly if the phrase *sees the man* exists in the translation model, and can thus be translated as one unit. Otherwise, if *the man* is translated out of context, we expect a wrong translation, since *der Mann* is more frequent than *den Mann* (the latter being the correct translation for this example).

With syntactic parsing, we can make the distinction between subjects and objects with sufficient accuracy, even over long distances.[4] From a syntactically enriched training corpus, the system can then learn distinct translations not only for every phrase, but also for every syntactic role of every phrase. Hence, the system no longer has to rely on contextual information for a correct translation of *the man*.

Ideally, the integration of parsing makes the SMT system more robust when the system has to fall back on short translating phrases. The baseline system can easily translate utterances such as *we support the report* or *the report is excellent*, since they are well covered in the training corpus.[5] Both *sees the man* and *sees the dog* are not seen in the training material, which means that the baseline system will most likely fail to produce a good translation.

### Method

A full list of Pro3Gres relations is reported in (Schneider 2008). Our main interest are the relations assigned to noun phrases, since we hope to predict the grammatical case of German noun phrases from the source text relations. The main relations for noun phrases in the Pro3Gres grammar are *subj*, *obj*, *obj2*, *modpp* and *pobj*. The last two are used for noun phrases embedded in prepositional phrases, either as modifiers (*modpp*) or verb complements (*pobj*). The first three are illustrated in examples 13 – 15.

13. The man *(subj)* sees the dog *(obj)*.

    Der Mann *(nom)* sieht den Hund *(acc)*.

14. The man *(subj)* gives the dog *(obj)* the ball *(obj2)*.

    Der Mann *(nom)* gibt dem Hund *(dat)* den Ball *(acc)*.

15. Peter *(subj)* is a man *(obj)*.

[4] About 84% F-value for subjects and direct objects in the English parser (Schneider 2008). These scores were obtained on a newspaper corpus, and performance on the spoken Europarl corpus is probably worse.

[5] *support the report* occurs appr. 200 times, *the report is* 650 times.

Peter *(nom)* ist ein Mann *(nom)*.

The relation label *obj* is used in three situations: for complements of transitive verbs (example 13), for the first complement of ditransitive verbs (example 14), and for noun predicates (example 15). We can see that there is no clear correspondence between the relation *obj* and case in the target language (German). This mismatch can be remedied by mapping the parser output to a new set of relations. By swapping *obj* and *obj2* in ditransitive verb clauses, *obj2* can be made to correlate with the dative case, *obj* to the accusative case. As a second transformation, all relations labeled *obj* that have a form of *to be* as their head are renamed *pred*. With these measures in place, we expect a sufficient correlation between the relation in English and German case. Naturally, not all ambiguities can be resolved this way. In German, the case of a prepositional noun may vary, depending on the preposition; the relation label *modpp* is hence of little use. Additionally, there are structural differences between the two languages, as we can see in example 16.

16. The commission *(subj)* is aware that [...]

    Der Kommission *(dat)* ist bekannt, dass [...]

The relation label *subj* typically corresponds to noun phrases in the nominative case. In example 16, the English subject instead corresponds to a German dative object. Such translations may be unexpected, but they do not pose a problem for SMT systems as long as they are consistent. The English phrase *one day* is translated as *eines Tages (gen)* in most instances, even if the parser assigns it the subject role. Similarly, *the day before yesterday* becomes *vorgestern* (an adverb) in German, no matter what syntactic role the English phrase is assigned. The only problem with example 16 is that the correct translations depends on local context, regardless of the syntactic role of *the commission*. This means that a specific model offers no advantage over the baseline system in cases similar to example 16.

The only system component affected in this experiment is the TM, which is computed with two factors on the source language side (surface form and relation), and one on

the target language side (surface form). This makes the factored TM more specific than the surface form TM used in the baseline. For the systems with alternative paths, the surface form TM is identical to that in the baseline. Word alignment, word reordering, and the language model are the same as in the baseline system.

### Results

The first observation we make is that we obtain scores that are significantly lower than the baseline ones when the specific translation model is used on its own. We expected this drop due to data sparseness; using alternative paths to combine the specific translation model with a general one, we can prevent a deterioration in score. However, the resulting system is not significantly different from the baseline system.

| Experiment | development set | | | test set | | |
|---|---|---|---|---|---|---|
| | BLEU | Unigrams | METEOR | BLEU | Unigrams | METEOR |
| baseline | 18.82 | 51.62 | 23.80 | 16.96 | 49.39 | 21.68 |
| factored only | 18.47 | 51.41 | 23.61 | **16.21** | 48.88 | 21.35 |
| alternative paths | 19.00 | 51.90 | 23.93 | 16.96 | 49.58 | 22.10 |

Table 5.3: Results for models with syntactic relation as input factor. Statistically significant ($p < 0.05$) differences from baseline BLEU score marked in bold.

We will now try to account for the lack of improvement. One might be tempted to say that mistranslations of grammatical case are too rare to have a significant effect on BLEU score. Indeed, the baseline translation of long sentences is so fragmented that it is difficult to establish if noun phrases are in the right case.

If it were true that real improvements exist and we are simply unable to measure them, we should be able to uncover them with short example sentences. Examples 11 and 12 were created with a special purpose in mind. The verb *sees* neither co-occurs with *the man* nor with *the dog* in the Europarl training corpus, which should impede the baseline system, but not the factored one. The translations of example 12 are shown in table 5.4. We can see that the system containing only the factored model produces

| source | the | dog | sees | the | man | . |
|---|---|---|---|---|---|---|
| reference | der | hund | sieht | den | mann | . |
| baseline | der hund | | sieht | der mann | | . |
| factored only | der hund | | sieht | den mann | | . |
| alternative paths | der hund | | sieht | der mann | | . |

Table 5.4: Different translations for example 12, segmented into individual phrase translations.

the best translation; both the system with alternative paths and the baseline system mistranslate *the man* in object position as *der mann*.

We offer the following interpretation of this finding. The specific model works as intended, producing the correct translation of example 12. Nevertheless, it is inferior to the general model, and consequently, the general (baseline) model is strongly preferred in the system with alternative paths.[6] We will continue our investigation as to why this is the case after discussing the second experiment with additional input factors.

### 5.3.2 Syntactic Head as Input Factor

#### Motivation

In the introductory example 1, we said that the correct translation of *stage* would be *Bühne*. This is true for that particular example, but *stage* has a wide range of meanings and has to be translated differently depending on its context. We will illustrate this with examples 17 and 18:

17. It means that we combat terrorism without tampering with fundamental freedoms and that the EU can *play an important part on the international stage*.

18. This seems unacceptable to me, not only because the programme for the social economy has yet to *begin its experimental stage* [...].

[6] This is not only visible in the similarity of the translations produced by the two systems, but also by looking at the model parameters themselves.

In example 17, the meaning of *stage* is the (metaphorical) place in which a performance takes place, translated as *Bühne*. In example 18, however, *stage* is a step of a process, corresponding to the German word *Phase*.

The problem is similar to that of word sense disambiguation (*WSD*). In the context of MT, we also speak of word translation disambiguation. Several methods of integrating WSD models into SMT systems have been investigated (Carpuat and Wu 2005; Carpuat and Wu 2007). Results were mixed, and Carpuat and Wu (2005) argue that phrase-based SMT systems are "sufficiently accurate, so that within the training domain, even the state-of-the-art dedicated WSD model is only able to improve on its lexical choice predictions in a relatively small proportion of cases." However, they did later manage to achieve significant improvements over a baseline SMT system (Carpuat and Wu 2007).

It is true that phrase-based SMT systems are good at disambiguating word translations, especially if the translation depends on local context. In example 17, local context does indeed help: *international stage* is a common phrase that occurs 228 times in the training section of the Europarl corpus, and it is most frequently translated as *internationale(n|r) Bühne*. Hence, the use of a phrase-based system increases the likelihood of a correct translation.

We can sabotage the phrase-based translation disambiguation by replacing *international* with *eurasian*. Since the phrase *eurasian stage* is not part of the translation model, a phrase-based SMT system will translate it no better than a word-based system. In isolation, *stage* is more commonly translated as *Phase*, which is the wrong word choice.

We propose to make word translation disambiguation more robust by adding the syntactic head of each token as an additional factor in the TM. The syntactic head is useful for word translation disambiguation, even if it is not part of the local context that is considered in a phrase-based system. Verbs like *play*, *act* or *perform* tend to co-occur with *stage* in the meaning of *Bühne*, while *begin*, *complete* or *reach* are indicative of *stage* as a step in a process. The same is true for articles and adjectives. Their inflection depends on the head noun, no matter if unseen adjectives such as *eurasian* are between the article and the noun.

## Method

From a theoretical standpoint, there are relations for which it is unclear which of the two tokens is the other's head.[7] Taking as an example the prepositional phrase, the argument can be made that the preposition is its head, since the preposition governs the inflection of the noun. However, choosing a content word (i.e. the noun) as head allows for a more direct access to the semantic structure of a sentence. For the Pro3Gres parser, Schneider (2008) defines the relation *modpp* as a relation between two content words, citing advantages for tasks such as Information Retrieval and Text Mining. Figure 5.1 shows the two alternatives. For SMT, both possible heads can provide valuable

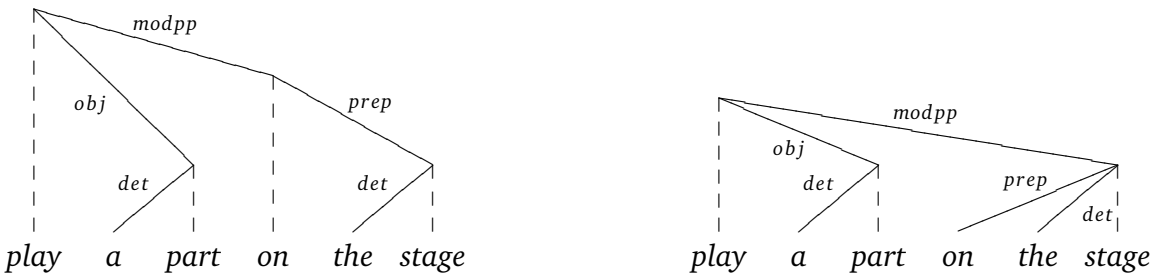


Figure 5.1: Different possible dependency trees of sample segment, depending on whether relation *prep* has preposition or noun as head.

information. On the one hand, knowing that *stage* depends on the preposition *on* is relevant for the inflection of the noun phrase in the German translation. On the other, the lexical head *play* helps in disambiguating between the different possible meanings of *stage*. We decided to use the heads that Pro3Gres provides. While the preceding preposition is also relevant to the translation of a noun phrase, the distance between preposition and noun phrase is usually short enough for phrase-based SMT systems to perform adequately well. Long-distance relations are more likely to provide information that a phrase-based SMT system cannot access otherwise.

[7] For an extended discussion on headedness, see (Schneider 2008).

For some constructions, the syntactic head provides little information. We used the ambiguity of adjectives as a motivational example for phrase-based SMT (see figure 2.1). If adjectives are used as predicates, word choice depends on the subject, which can be several words away. This means that a phrase-based approach is not able to take the subject into account for the translation of the predicative adjective. Figure 5.2 shows the dependency tree of example 19.

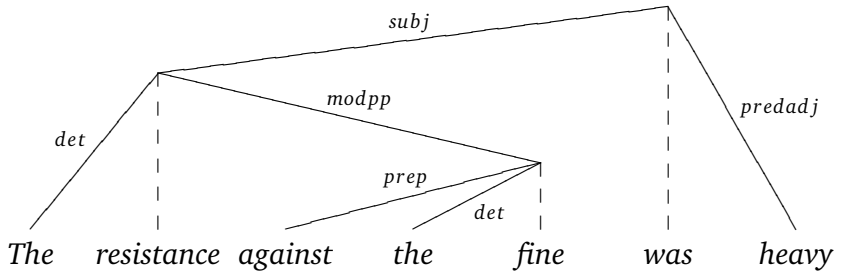


Figure 5.2: Pro3Gres dependency tree of a sentence with a predicative adjective.

19. The resistance against the fine was heavy.

The syntactic head of *heavy* is the root verb *was*, which does not help the disambiguation between the possible translation candidates *schwer*, *dicht*, *heftig* or *hoch* that we have seen in figure 2.1. Furthermore, the immediate context of *heavy* is not relevant to finding the right translation. To the contrary, the phrase pair (*the fine was heavy*|*die Buße war hoch*) might even exist in the TM and lead to the wrong translation *der Widerstand gegen die Buße war* ***hoch*** instead of *der Widerstand gegen die Buße war* ***heftig***. With access to the full parser output, we can formulate a rule that does not return the actual syntactic head as additional factor of predicate adjectives, but their subject. Thus, we provide the model with more useful information.

When adding syntactic heads as factors, data sparseness is a major issue, as we have seen in table 4.1. We use the lemmas of the heads to mitigate this problem. Still, we have to combine the specific TM with a general one to keep the number of unseen

tokens low. The experimental systems are identical to the baseline system except for the factored TM, which has the syntactic head as additional input factor, and MERT.

### Results

All results are shown in table 5.5. As expected, the specific model performs significantly worse than the general model when used on its own. We attribute this to the high num-

| Experiment | development set | | | test set | | |
|---|---|---|---|---|---|---|
| | BLEU | Unigrams | METEOR | BLEU | Unigrams | METEOR |
| baseline | 18.82 | 51.62 | 23.80 | 16.96 | 49.39 | 21.68 |
| factored only | **15.19** | 47.14 | 19.74 | **13.31** | 44.46 | 17.72 |
| alternative paths | 19.00 | 51.86 | 24.03 | 16.81 | 49.30 | 21.82 |

Table 5.5: Results for models with syntactic head as input factor. Statistically significant ($p < 0.05$) differences from baseline BLEU score marked in bold.

ber of token/factor combinations that are not seen during training, and the low number of occurrences of those combinations that do occur in the training material (relative to the surface form). We observed no statistically significant difference between the results of the baseline system and those of the system with alternative paths.

We selected example 19 for a sandbox experiment, since it is one where we expect the phrase-based system to fail. We have already discussed that phrase-based systems are good at word translation disambiguation if it can be performed on the basis of local context. This includes *heavy resistance*, but not *the resistance against [...] was heavy*.

| | |
|---|---|
| source | the resistance against the fine was heavy |
| reference | der widerstand gegen die buße war heftig |
| baseline | der widerstand gegen die geldstrafe war schwere |
| factored only | der widerstand against fine war die heftigen |
| alternative paths | der widerstand gegen die geldstrafe war schwere |

Table 5.6: Different translations for example 19. Experiments add syntactic head.

The baseline and experimental translations are shown in table 5.6. We can see that

the factored model does a better job at translating *heavy*, although the form of *heftig* is wrongly inflected, so the translation would still be considered wrong by BLEU. Still, the translation is preferable to *schweren*, which is not only an unusual (and hence improbable) word choice as an adjective to *Widerstand*, but also has the wrong inflection. The second difference we spot is that the factored model leaves *against* and *fine* untranslated, apparently because *against*|*resistance* and *fine*|*resistance* do not occur in the training corpus. This confirms our expectation that the number of unseen/untranslated tokens is higher in the factored model than in the general one, as we can deduct from table 4.1.

The translation of the system with alternative paths is identical to that of the baseline system. As in our last experiment, the system with alternative paths strongly prefers the general TM.

### 5.3.3 Problem Analysis

The root of most problems we encountered is data sparseness. In our experiments, data is significantly sparser for the training of the factored TMs. This has a number of implications that partly explain why our experiments were unsuccessful.

The most apparent effect of data sparseness is the high number of token/factor combinations in the test set that are not seen during training. In fact, not only unseen tokens are relevant, but also longer phrases that are not seen during training. If a source phrase is unseen, it is translated by segmenting it into smaller phrases (or tokens) and translating these individually. How this can negatively affect performance is shown in table 5.7. The source sentence can only be translated correctly if *the day before yesterday* is translated as single unit, as the baseline system does. However, the factored phrase *the*|*day day*|*speak before*|*yesterday yesterday*|*day* does not occur in the training corpus. The higher number of unseen tokens and phrases in the factored TM necessitates its combination with a general TM. What we will try to answer now is why the factored TM was heavily disfavoured in the log-linear combination through MERT.

| source | he | spoke | the | day | before | yesterday | . |
|---|---|---|---|---|---|---|---|
| reference | er | sprach | vorgestern | | | | . |
| baseline | er sprach | | vorgestern . | | | | |
| factored only | er hat | | der | tag | vor | gestern | . |
| alternative paths | er sprach | | vorgestern . | | | | |

Table 5.7: Different translations for *he spoke the day before yesterday*, segmented into individual phrase translations.

We have stated earlier that MLE is unreliable when the number of observations of a source phrase is low. To investigate this hypothesis, we extracted all phrase pairs with

| source phrase | target phrase | $\phi(\bar{s}\|\bar{t})$ | $lex(\bar{s}\|\bar{t})$ | $\phi(\bar{t}\|\bar{s})$ | $lex(\bar{t}\|\bar{s})$ |
|---|---|---|---|---|---|
| dog | hund | 0.727 | 0.444 | **0.364** | 0.255 |
| dog | katzenfutter | 1 | 1 | **0.045** | 0.021 |
| dog | gesetzes über hunde- | 1 | 0.057 | **0.045** | 9.631e-06 |
| dog | man | 3.854e-05 | 3.560e-05 | **0.045** | 0.021 |

Table 5.8: Phrase translations of *dog* in the surface form TM (excerpt).

*dog* as source phrase, both from the surface form TM (table 5.8) and the factored one (table 5.9). We will focus the discussion on the phrase translation probability $\phi(\bar{t}|\bar{s})$, one of the five translation probabilities provided[8], since it is easiest to understand without context[9], and since the sum of all probabilities $\phi(\bar{t}|\bar{s})$ is 1 for any phrase $\bar{s}$.

What we find out is that MLE for *hund* is based on 22 phrase pairs, and that 14 of them are noise.[10] Since the phrase pair $(hund|dog)$ occurs 8 times, and each of the noisy pairs only once, the good phrase pair is estimated to be 8 times more probable than each of the noisy ones. This number increases dramatically when the number of observations is large. As translation for *report*, *bericht* is 570 times more probable than *er* (the best-scoring phrase pair we consider noisy), and about 10000 times more

[8] Only four are shown in tables 5.8 and 5.9, the fifth being a constant phrase penalty.

[9] $\phi(\bar{s}|\bar{t})$ has to be considered in relation to the LM probablity and is uninformative on its own.

[10] The number of occurrences of a phrase in the training corpus is not identical to the number of valid phrase pairs extracted through phrase alignment, owing to the phrase alignment heuristic used by (Koehn, Och and Marcu 2003).

| source phrase | target phrase | $\phi(\bar{s}\|\bar{t})$ | $lex(\bar{s}\|\bar{t})$ | $\phi(\bar{t}\|\bar{s})$ | $lex(\bar{t}\|\bar{s})$ |
|---|---|---|---|---|---|
| dog\|view | hund | 0.091 | 0.037 | **1** | 1 |
| dog\|factory | katzenfutter | 1 | 1 | **1** | 1 |
| dog\|act | gesetzes über hunde- | 1 | 0.057 | **1** | 0.037 |
| dog\|drown | man | 3.854e-05 | 3.560e-05 | **1** | 1 |

Table 5.9: Phrase translations of *dog* in the factored TM (excerpt).

probable than *darin*, among others the worst-scoring target phrase with probability $>$ 0. These probability estimations are based on about 14500 observations of *report*. As a rule, the more often a source phrase occurs, the more improbable we expect noisy translations to become (compared to good ones). This impressively shows how a large number of observations reduces the impact of noise on phrase translations.

In contrast, noise is a major problem in the factored TM. Most factored strings formed with *dog* only occur once, so that the whole probability space can be taken up by a noisy translation. This is the case for *dog* with *act*, *drown* or *factory* as its head. We are unable to estimate accurate translation probabilities from single observations; accordingly, the factored model often overestimates the probability of bad translations. It comes as no surprise then that the surface form model is preferred by MERT.

While the phrase translation probability $\phi(\bar{t}|\bar{s})$ is unreliable for rare phrase pairs, $\phi(\bar{s}|\bar{t})$ creates a more general problem in the factored TM. $\phi(\bar{s}|\bar{t})$ can be several orders of magnitude lower than the probabilities in the inverse direction.[11] Paradoxically, the more frequent a source word is, or more precisely, the more different factors it co-occurs with, the more heavily it is penalised in the factored model. For the phrase pairs occurring only once in tables 5.8 and 5.9, $\phi(\bar{s}|\bar{t})$ is unaffected. However, if we look at a more frequent source phrase such as *report*, the effect is tremendous. Consider table 5.10. We can see that phrase pairs in the factored model have values for the phrase translation probability $\phi(\bar{s}|\bar{t})$ and the lexical weight $lex(\bar{s}|\bar{t})$ that are several magnitudes

[11] We can understand this phenomenon by observing the effect factored input has on our MLE estimation (equation 6, page 26). It does not affect the denominator, but the average numerator is $n$ times smaller than in the general model, $n$ being the number of different factors existing for the same surface form phrase.

| source phrase | target phrase | $\phi(\bar{s}\|\bar{t})$ | $lex(\bar{s}\|\bar{t})$ | $\phi(\bar{t}\|\bar{s})$ | $lex(\bar{t}\|\bar{s})$ |
|---|---|---|---|---|---|
| report (no factors) | bericht | 0.861 | 0.930 | 0.737 | 0.762 |
| report\|announce | bericht | 0.00005 | 0.00005 | 1 | 1 |
| report\|write | bericht | 0.002 | 0.002 | 0.720 | 0.787 |
| a report (no factors) | einen bericht | 0.695 | 0.485 | 0.428 | 0.056 |
| a\|report report\|announce | einen bericht | 0.001 | 0.000001 | 1 | 0.427 |
| a\|report report\|write | einen bericht | 0.013 | 0.00007 | 0.462 | 0.336 |

Table 5.10: Selected phrase pairs in the factored and surface form TM.

lower than in the surface form model. For longer phrases, $\phi(\bar{s}|\bar{t})$ tends to increase again, while $lex(\bar{s}|\bar{t})$ continues to decrease, being a multiple of low probabilities.

When trying to translate a single phrase, low probabilities are not necessarily problematic; since all translation options will have similarly low probabilities, calculating the most likely translation will still give generally good results. However, we will face problems when different phrase segmentations and/or different translation models compete for producing the translation with the highest probability. We find it hard to imagine how the weights would have to be picked to properly balance the probability of short and long phrases on the one hand, the factored and surface form TM on the other hand.

In summary, we have not been able to use alternative paths to combine the two translation models in a way so that overall performance exceeds that of the baseline. We still maintain that the factored model is beneficial under the right circumstances, though we are unable to selectively use the factored TM at the right time with log-linear modelling, which means that systems with alternative paths mostly ignore the factored TM.

### 5.3.4 Investigating Better Ways to Combine Translation Models

Seeing that a log-linear combination of the models did not produce the desired results, we inspected alternative ways of integrating the factored TM into our baseline system. We found that the factored model is given bad weights and thus mostly ignored in the log-linear system, even though our intuitive assumption is that the factored TM is

useful in specific cases. We are also confident that we can formulate rules as to when the factored TM should be preferred.

In an initial experiment, two methods of filtering the factored translation model have been investigated. Most importantly, we eliminated source phrases observed fewer than 10 times during training ($count(\bar{s}) < 10$) from the factored TM. We saw that the quality of probabilities estimated by MLE deteriorates if $count(\bar{s})$ is low, and we want to disregard the factored model in those cases.[12] Also, we test only considering NP relations in our factored model, since we expect these to be most informative for our purpose. The resulting TM will contain fewer phrase pairs, but these should be of higher quality and produce better results than those in the surface form TM.

To get even more control over the combination of translation models, we implemented a deterministic back-off chain, analogous to Katz's back-off (Katz 1987), as an alternative to a log-linear combination of models. If an $n$-gram is rare or unseen in a training corpus, Katz proposes to recursively utilize shorter $n$-grams to estimate its probability. Similarly, we have implemented a model that backs-off to the surface form model if a factored source phrase is rare. Potentially, we can determine arbitrary conditions to control whether translation options are extracted from the factored TM or its back-off, the surface form TM. The decision is made for every phrase $\bar{s}_i^j$, that is, a sequence of words starting at position $i$ and ending at position $j$ of a sentence with $n$ words ($1 \leq i \leq j \leq n$). In the system with alternative paths, translation options of each phrase are extracted from both TMs and compete for generating a translation with the highest probability. With a deterministic back-off, we combine a number of TMs with precedence rules, extracting translation options from the TM with the highest precedence if possible, and backing-off to the TM of next-lower precedence until the input phrase is found. We call this back-off deterministic since only one TM provides the translation options for each phrase, and the others are ignored altogether. By fil-

[12] It is not always ideal to filter the factored TM. When working with an analytical decoding path that translates lemmas, we expect the surface form model to be sparser. In this case, the surface form model should best be filtered.

tering the individual translation tables, we can formulate complex decision rules, such as extracting the translation options from the factored model if $count(\bar{s}) \geq 10$ and if the source phrase factors are all NP relations, and using the back-off model otherwise. With these rules, we force the decoder to use the factored TM as often as possible, but back-off to the surface form TM when data in the factored TM is too sparse.

As a second modification, we decided to replace $\phi(\bar{s}|\bar{t})$ and $lex(\bar{s}|\bar{t})$ in the factored TM with the values from the surface form model. Besides avoiding the problems with the probabilities illustrated in the last section, this also allows us to use the same model parameters for all TMs. Giving every model separate weights would be harder to implement and has the serious drawback that MERT takes several times longer and produces results that are less reliable and more prone to overfitting. This would become especially obstructive when implementing more than one back-off level.

We will evaluate whether taking $\phi(\bar{s}|\bar{t})$ and $lex(\bar{s}|\bar{t})$ from the surface form TM does indeed improve performance. We will also compare different decision rules to see if the results confirm our hypothesis, namely that we can produce better results by disallowing certain phrase pairs.

### Results

As to the question whether it makes sense to discard $\phi(\bar{s}|\bar{t})$ and $lex(\bar{s}|\bar{t})$ from the factored models, using the probabilities from the surface form model instead, we provide tentative results in table 5.11. Either all probabilities in the factored model are original (as estimated by MLE), or mixed, meaning that $\phi(\bar{s}|\bar{t})$ and $lex(\bar{s}|\bar{t})$ are from the baseline model, $\phi(\bar{t}|\bar{s})$ and $lex(\bar{t}|\bar{s})$ from the factored one. We observe that the difference between the two systems is not statistically significant. Since we deem the original probabilities more problematic when trying to combine several TMs, we still opt to continue our experiments with mixed probabilities. This has the additional advantage that all experiments can be conducted with the same model parameters.

| Experiment | development set | | | test set | | |
|---|---|---|---|---|---|---|
| | BLEU | Unigrams | METEOR | BLEU | Unigrams | METEOR |
| original prob. | 15.19 | 47.14 | 19.74 | 13.31 | 44.46 | 17.72 |
| mixed prob. | 15.24 | 47.30 | 20.00 | 13.48 | 45.06 | 18.08 |

Table 5.11: Results for model with syntactic head as input factor. Statistically significant ($p < 0.05$) differences in BLEU score marked in bold.

Having confirmed that mixed probabilities are viable, we can now investigate deterministic back-offs. For all experiments with deterministic back-offs, we used the model parameters of the baseline system. New training of the factored model did not lead to significantly better results, and using the same parameters eliminates a confounding factor, thus making the results more comparable. Table 5.12 shows results obtained using different sets of rules to decide when to use the factored model. All factored systems have mixed probabilities ($\phi(\bar{s}|\bar{t})$ and $lex(\bar{s}|\bar{t})$ are from the baseline model, $\phi(\bar{t}|\bar{s})$ and $lex(\bar{t}|\bar{s})$ from the factored one). In order to better verify whether improvements are statistically significant, we conducted these experiments on a larger test set of 2000 sentences. The exact setting of each experiment is as follows:

| Experiment | BLEU | Unigrams | METEOR |
|---|---|---|---|
| baseline | 19.39 | 51.58 | 24.25 |
| rel | 19.46 | 51.81 | 24.39 |
| relMin10 | **19.60** | 51.88 | 24.50 |
| relMin10NP | 19.45 | 51.70 | 24.33 |
| head | 19.51 | 51.65 | 24.30 |
| headMin10 | **19.62** | 51.79 | 24.44 |
| headRelMin10 | **19.66** | 51.84 | 24.48 |
| headRelMin10Mult | **19.75** | 52.08 | 24.57 |

Table 5.12: Results for model with additional input factors, using a back-off system. Statistically significant ($p < 0.05$) differences from baseline BLEU score marked in bold.

**baseline** surface form model; no back-off

**rel** relation as input factor; baseline model as back-off

**relMin10** relation as input factor; $count(\bar{s}) \geq 10$; baseline model as back-off

**relMin10NP** relation as input factor; $count(\bar{s}) \geq 10$; only NP relations; baseline model as back-off

**head** head as input factor; baseline model as back-off

**headMin10** head as input factor; $count(\bar{s}) \geq 10$; baseline model as back-off

**headRelMin10** head + relation as input factors; $count(\bar{s}) \geq 10$; baseline model as back-off

**headRelMin10Mult** head + relation as input factors; $count(\bar{s}) \geq 10$; models from experiments headMin10, relMin10 and baseline as back-offs (in this order)

Four of the seven systems achieve significantly better BLEU scores than the baseline system, the difference being up to 0.36 BLEU points. We were surprised to find that including rare source phrases in the factored TM did not lead to a drop in performance. Still, we could demonstrate that the systems which only include more frequent source phrases yield better scores. With 0.1 BLEU points, the difference seems small. On closer inspection, we see that precision actually decreases in the systems whose factored TMs contain rare source phrases. Only because the system translations are longer, which reduces the brevity penalty, are BLEU scores slightly (but not significantly) higher than the baseline scores. Disregarding the brevity penalty, unigram precision is 0.35 points higher in the system *headMin10* than in *head*, and 0.3 points higher in *relMin10* than in *rel*.

We have investigated more closely what threshold for $min(count(\bar{s}))$ is optimal. Figure 5.3 shows that most scores improve up to a threshold value of 6, and then start to slowly converge towards the baseline score, which will be reached when the threshold is so high that all phrase pairs from the factored TM are discarded. It is unclear in how far these findings can be generalised, that is if the optimal threshold is constant for smaller or larger training corpora, different factors, model parameters and test sets.

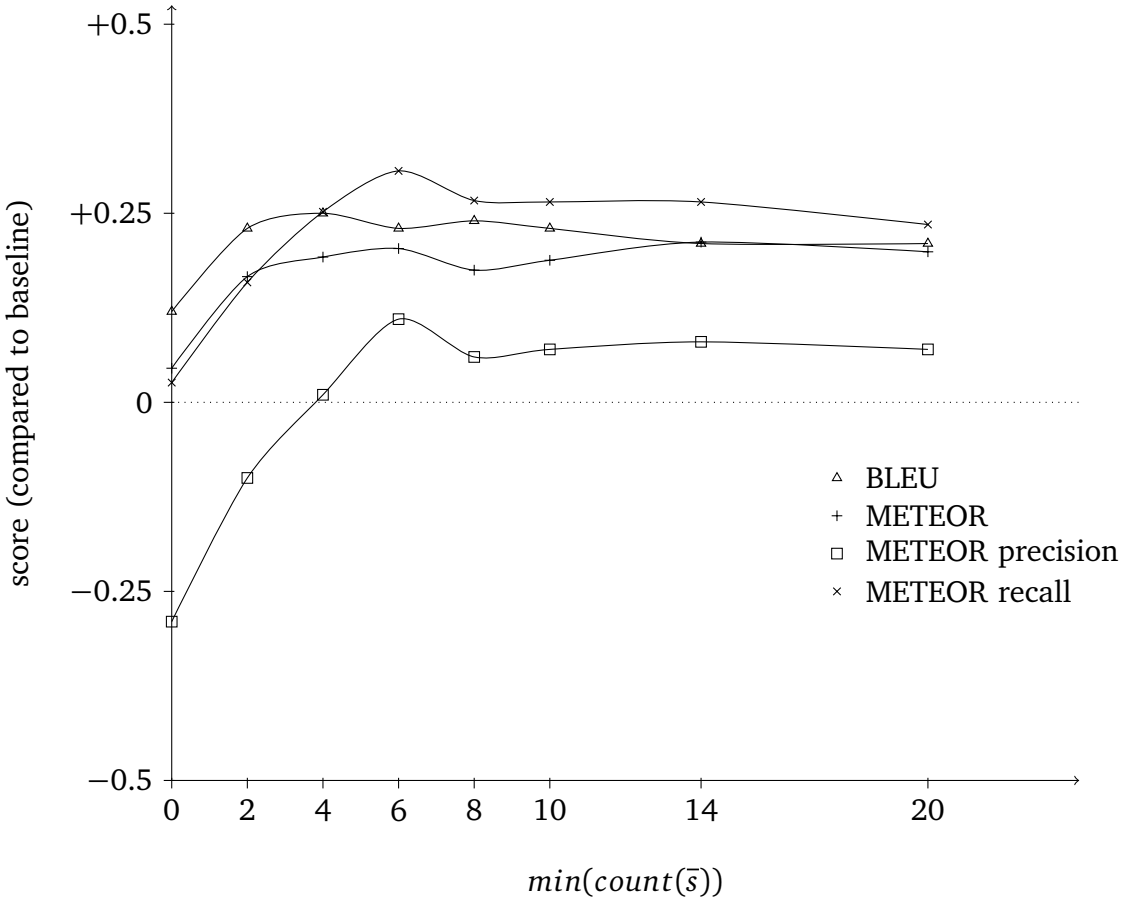


Figure 5.3: Scores obtained with different thresholds for minimum number of occurrences of source phrase during training. Source phrases below the threshold are discarded from the factored TM. Factor is syntactic head.

Going back to the results in table 5.12, only allowing NP relations as factors reduced the score gain. This indicates that not only NP relations, but also other relations help to disambiguate the translation of ambiguous surface forms. We can see that system performance benefits from multiple back-offs. *headRelMin10* and *headRelMin10Mult* are identical with the exception that the former backs-off directly to the baseline model, while the latter has three back-off levels. Using multiple back-offs yields an improvement of 0.1 BLEU points over the system with only one back-off level; filtering rare source phrases an improvement of 0.1-0.15 BLEU points. While the benefit of each measure is small, the best system outperforms the baseline system by 0.35 BLEU points.

We have included five translations that differ between the baseline and the best-performing experimental system (*headRelMin10Mult*) in the appendix (table I.1, page 101). Of those five translations, the last shows the clearest improvement of the experimental system. The English token *a.m.* is translated out of context, since it does not co-occur with *10.56* in the training corpus. Consequently, the baseline system mistranslates it as *Uhr wieder aufgenommen*, while the experimental system receives the information from the parser that the syntactic head of *a.m.* is *close*, and thus produces a correct translation.[13]

Considering both the automatically obtained results and an investigation of sample translations, we consider this experiment a mild success. We would not recommend this approach in a productive environment, given the additional effort required by parsing, but we could show that an improvement is possible, even though initial experiments suggested otherwise.

### 5.3.5 Alternative Paths versus Deterministic Back-offs

In the preceding section, we report an improvement using deterministic back-offs. Some of our modifications can also be applied to a system with alternative paths, and we will now try to determine in how far this is viable.

Table 5.13 shows the results obtained by combining filtered TMs and the surface form TM using alternative paths. Restricting the factored TM to NP relations led to a significant improvement on the development set, but not on the test set. We observe the same phenomenon when only allowing source phrases that occur at least 10 times in the training corpus: development set scores are significantly higher, but not the test set ones. We attribute this phenomenon to overfitting. We have not attempted to use a larger development set for tuning, since MERT has proven very time-consuming. Instead, we continued experiments with mixed probabilities and model parameters of the baseline.

[13] From a linguistic point of view, it is unsatisfactory that *was closed* is translated as *wird*, and *a.m.* as *Uhr geschlossen*. We would prefer a correct alignment and subsequent reordering, but in the end, we achieved our goal of producing a correct translation.

By using the same model parameters for all TMs, we avoid the need for tuning the up to 30 parameters per system.

| Experiment | development set | | | test set | | |
|---|---|---|---|---|---|---|
| | BLEU | Unigrams | METEOR | BLEU | Unigrams | METEOR |
| baseline | 18.82 | 51.62 | 23.80 | 16.96 | 49.39 | 21.68 |
| relNP | **19.20** | 51.68 | 24.14 | 16.76 | 49.12 | 21.86 |
| relMin10 | **19.52** | 52.18 | 24.41 | 16.89 | 49.54 | 21.95 |
| headMin10 | **19.17** | 51.94 | 24.09 | 16.89 | 49.46 | 22.07 |

Table 5.13: Results for filtered translation models. Statistically significant ($p < 0.05$) differences from baseline BLEU score marked in bold.

The results obtained with these systems are shown in table 5.14 The systems are mostly identical to those described in the last section, and will thus be directly compared to them. The difference between the two approaches is that the decoder may use translation options from the baseline model if the probabilities warrant it, even if the factored TM contains translation options for the same phrase. In the backed-off system, only translation options from the factored model are considered, if they exist.

The results are inconclusive as to whether systems with deterministic back-offs or alternative decoding paths perform better, although both approaches significantly outperform the baseline system. Only in the systems with four translation paths does backing-off offer a slight, albeit not statistically significant, advantage over the combination through alternative paths. The most apparent difference lies in decoding time. Systems with alternative paths are up to 10 times slower than the backed-off ones, since more hypotheses are built up and scored.

We conclude that a deterministic back-off is not absolutely necessary to successfully combine up to four different translation models. Nor is any of the other measures taken indispensable, these being the discarding of rare phrase pairs in the factored TM, mixed probabilities, and avoiding MERT by using baseline parameters for all TMs. However, all of them led to small gains in performance, which in the end allowed the combined systems to outperform the baseline system. We also want to highlight that

| Experiment | deterministic back-offs | | | alternative paths | | |
|---|---|---|---|---|---|---|
| | BLEU | Unigrams | METEOR | BLEU | Unigrams | METEOR |
| baseline | 19.39 | 51.58 | 24.25 | 19.39 | 51.58 | 24.25 |
| relMin10 | **19.60** | 51.88 | 24.50 | **19.58** | 51.83 | 24.42 |
| headMin10 | **19.62** | 51.79 | 24.44 | **19.61** | 51.81 | 24.41 |
| headRelMin10 | **19.66** | 51.84 | 24.48 | **19.66** | 51.85 | 24.47 |
| headRelMin10Mult | **19.75** | 52.08 | 24.57 | **19.68** | 52.01 | 24.53 |

Table 5.14: Comparison between deterministic back-offs and alternative paths as methods to combine TMs. Statistically significant ($p < 0.05$) differences from baseline BLEU score marked in bold.

using the same model parameters for all experiments results in a high comparability of the systems. We can be certain that the improvement we observed is indeed caused by the syntactically enriched translation models, and not by other confounding factors.

## 5.4 Syntactic Relation as Output Factor

### 5.4.1 Motivation

It has already been demonstrated that adding a POS language model improves translation quality (Koehn and Hoang 2007). We will illustrate the motivation for doing so with an example: Let us assume that we want to estimate which word sequence is more likely: *der kleine Mann schläft* or *Mann kleine der schläft*. Let us further assume that the language model has not seen any of the bigrams in either of the candidates due to data sparseness. As a consequence, both candidates will receive the same score from the language model. The POS representation of the first candidate is *ART ADJA NN VVFIN*, the one of the second *NN ADJA ART VVFIN*. Even without knowing the actual word forms, a competent German speaker will be able to say that the first POS sequence is far more frequent than the second. Also, we have better chances to see this sequence in a training corpus: Even though neither of the word sequences occurs in the Europarl training set, *ART ADJA NN VVFIN* does so 37,000 times; *NN ADJA ART VVFIN* does not. These numbers nicely illustrate that data sparseness is a far smaller problem on a POS-level. This

means that larger *n*-grams can be evaluated by the language model so that an unnatural word order is penalized more heavily.

The question is if the dependency labels provided by syntactic parsing can perform the same function equally well or even better. In many ways, dependency labels are similar to POS tags: First of all, both analyses are similarly fine-grained[14]: the words are classified into 54 classes by POS tagging, and into 41 by parsing. In some cases, there is an almost one-to-one correspondence between word class and syntactic functions, as for articles. POS tagging is more fine-grained in the classification of verbs, using 12 classes for different verb forms. In contrast, dependency parsing assigns one of 8 different labels to a noun, according to its syntactic role[15]. Since roles are marked by grammatical case, dependency labels serve as a partial morphological analysis of a text. Grammatical number and gender is not determined in Pro3Gres parsing, although a full morphological analysis was used in (Koehn and Hoang 2007). We will see if the lack of a full morphological analysis is a deficiency, or if we can reach the amount of improvement cited in (Koehn and Hoang 2007).

### 5.4.2 Method

In our experiment, the system will produce factored output which is then scored on two language models: one language model evaluates the surface form sequence, another the sequence of syntactic relations. The German Pro3Gres parser knows 29 relations, its grammar being based on (Foth 2005), who provides a full list of the relations.

The main advantage of language models on syntactic relations is that data sparseness is not a problem when we have only 29 types and a 30-million-words training corpus. This allows us to increase the number of relations without detrimental effects. Koehn and Hoang (2007) have used a language model over morphological factors to increase intra-NP agreement. Pro3Gres does not provide a full morphological analysis, but case

[14]The following statements are based on the Pro3Gres grammar, which will be discussed in more detail in the next section, and the STTS tagset (Schiller et al. 1999).

[15]These being *subject*, *accusative object*, *dative object*, *genitive object*, *genitive modifier*, *predicate noun*, *apposition* and *temporal modifier*.

information can be inferred from the grammatical function of nouns. We have modified the *determiner* and *attribute adjective* relations to include the grammatical function of their head. Furthermore, the syntactic relations are underspecified at points. There is no difference in the parser output between finite verbs that are the root of a sentence, tokens that are not attached to any head due to parsing errors, and punctuation marks. We added mapping rules that distinguish between these phenomena and use different relation labels for each. This allows for a more discriminative language model, as figure 5.4 shows. A language model trained on the unmodified relation labels will assign the

| surface forms (f) | relations (f) |
|---|---|
| den mann , der hund (0) | detobja subj comma detsubj obja (0) |
| der mann sieht den hund (0) | detsubj subj finverb detobja obja (10.511) |

Figure 5.4: Hypothetical translation output for example 13, and frequency of each sequence in the training corpus.

same probability to the two sentences *den mann , der hund* and *der mann sieht den hund*, since both are expressed by the same relation sequence *det subj root det obja*. After the mapping, the grammatically correct sequence receives a higher probability from the LM.[16] The final number of relations used is 41.

The experimental system is different from the baseline in two aspects. Firstly, an additional LM is used that works on German syntactic relations. Secondly, the TM is factored on the target language side. Having factors only on the target language side has the advantage that parsing is only needed during training, and not during decoding.

### 5.4.3 Results

As table 5.15 shows, the experimental model achieves a significant improvement of 0.4 points BLEU score on the development set, but no significant improvement on the test set. Looking at the development set, the BLEU unigram score does not increase as

[16] It is not necessary that the sequence *der mann sieht den hund* is translated with the right relation labels. Another possibility is that the surface forms in the translation are correct, but the relation labels wrong.

strongly as the full BLEU score, which indicates that fluency, not adequacy, is increased by the additional LM. This is consistent with our expectation.

| Experiment | development set | | | test set | | |
|---|---|---|---|---|---|---|
| | BLEU | Unigrams | METEOR | BLEU | Unigrams | METEOR |
| Baseline | 18.82 | 51.62 | 23.80 | 16.96 | 49.39 | 21.68 |
| Factored output/LM | **19.21** | 51.72 | 24.13 | 16.99 | 49.13 | 22.12 |

Table 5.15: Results for models with syntactic relation as output factor and in LM. Statistically significant ($p < 0.05$) differences from baseline BLEU score marked in bold.

Table 5.16 shows the translation of a sample sentence. For this sentence, the experimental system fails to produce a better translation than the baseline. The problem is that *die*, while theoretically meeting case constraints, is either the wrong gender or number for *mann*. Five randomly selected translations that differ between the baseline and the experimental system are included in the appendix (table I.2).

| source | the | dog | sees | the | man | . |
|---|---|---|---|---|---|---|
| reference | der | hund | sieht | den | mann | . |
| baseline | der hund | | die | | menschen . | |
| baseline 2 | der hund | | sieht | der mann | | . |
| factored output | der hund | | betrachtet die | | mann | . |

Table 5.16: Segmentation of system translations into individual phrase translations.

The improvements reported in (Koehn and Hoang 2007) are small for the language pair English–German: 0.2 BLEU score points. It is well possible, given better luck with MERT, that we could have obtained similar results. On the other hand, our implementation was different in that we did not work with POS tags and morphological tags, but with syntactic relations. We did not pursue this approach further since we are unlikely to gain insights beyond what is already known, especially since Pro3Gres relations are inadequate to enforce agreement within noun phrases.

## 5.5 Reordered Models

### 5.5.1 Motivation

We have already seen that discontiguous phrases in German pose a major problem for translation (see example 6 (page 29)). Source language reordering in preprocessing mitigates this problem, and there has been successful research in this field (Nießen and Ney 2004; Collins, Koehn and Kučerová 2005; Popović and Ney 2006; Holmqvist et al. 2009). However, not all reordering problems can be solved by preprocessing the source language.

If the translation direction in example 6 (page 29) is German–English, the source phrase can be reordered, but not if it is English–German. A typical result of translating with the baseline system is that the verb particle is lost in the translation. This may seem minor and will not have a big influence on automatic measures of translation quality, but the verb phrase is semantically very important. Incorrectly translating *propose* to *schlagen* changes the meaning of the sentence to *We beat a temporary Commission*.

The usefulness of source language reordering has been well documented; we will not try to reproduce earlier findings, but to investigate the feasibility and effect of target language reordering.

### 5.5.2 Method

We have investigated two reordering algorithms. Our experiment with full reordering is based on the heuristic described in (Collins, Koehn and Kučerová 2005). This heuristic aims at reordering the German sentence so that word order is as parallel as possible to its English counterpart. This includes always placing the subject directly before the verb and putting all verbs in the verbal complex directly afterwards, even in subordinate clauses, in which German verbs normally occur in clause-final position. For a second experiment, only German verbal particles, which we identified as a phenomenon that cannot be reordered in the English source text, are placed immediately after the verb.

All models (translation model, reordering model and language model) are trained on the reordered German corpus. As a result, our initial translation will ideally be *Wir schlagen vor eine provisorische Kommission* for example 6. In other words, our system only produces reordered German sentences. We train a second Moses model to translate from reordered German sentences to un-reordered (i.e. original) ones. We thus decompose translation into two distinct statistical translation steps (three if we count the recaser).

### 5.5.3 Results

We obtained the best results by using the same model parameters as in the baseline experiment, and hence report only test set scores. Table 5.17 shows the results of the reordering experiments. We report scores after the initial translation step and after the reordering step, both for full reordering and verbal particle only reordering. We can thus observe how effective our second translation step is in restoring the correct word order.

| Experiment | BLEU | Unigrams | METEOR |
|---|---|---|---|
| baseline | 16.96 | 49.39 | 21.68 |
| full; 1st step | **16.03** | 49.75 | 21.18 |
| full; final | **16.21** | 49.73 | 21.31 |
| verb part. only; 1st step | 16.91 | 49.41 | 21.83 |
| verb part. only; final | 16.99 | 49.40 | 21.88 |

Table 5.17: Results for reordered models. Statistically significant ($p < 0.05$) differences from baseline BLEU score marked in bold.

The full reordering system performs significantly worse than the baseline system. Interestingly, the unigram scores are better, which means that the lower scores can be fully attributed to a deterioration in *n*-gram scores of higher order. The second translation step can remedy some of the word order problems of the first translation step, but the final system is far from reaching the scores of the baseline system, the difference being 0.75 BLEU point. One sentence taken from the Europarl corpus shows

the typical differences between the baseline and the reordered system (table 5.18). Both

| | |
|---|---|
| source | [...], not because they are less important,<br>but because I wanted to focus on them for a moment. |
| reference | [...], nicht weil sie am unbedeutendsten *sind*,<br>sondern weil ich einen Moment bei ihnen *verweilen wollte*. |
| baseline | [...], nicht, weil sie weniger wichtig *sind*,<br>sondern weil ich hierauf zu *konzentrieren*. |
| full; 1st step | [...], nicht weil sie *sind* weniger wichtig,<br>sondern weil ich *möchte eingehen* kurz auf sie. |
| full; final | [...], nicht weil sie *sind* weniger wichtig,<br>sondern weil ich *möchte* kurz auf sie *eingehen*. |

Table 5.18: Sample translation for system with full target language reordering. Verbs emphasised

clauses shown[17] are subordinate clauses, which means that the inflected verbs (*sind* and *möchte/wollte*, respectively) should be clause-final. We can see that the reordered model does not correctly restore verb-final word order, but V2 order typical for main clauses. The baseline system translates the first clause correctly, but lacks a main verb in the second one. This supports our interpretation of the numeric results. Unigram precision is slightly better in the reordered model, but word order is worse.

By looking at more translations of the reordered model, we propose two explanations for the second translation step failing to reverse our initial reordering and restore the correct word order. Firstly, the second translation step suffers from the same shortcomings as the baseline system. The word and phrase alignment algorithms are unable to deal with discontiguous phrases and can thus only correctly reorder over a short distance. Secondly, we throw away the distinction between main clauses and subordinate clauses in the reordered model, and converting a reordered verb phrase to a clause-final one is often impossible for the statistical model. A rule-based preprocessing of the source phrase is far better at performing the long-distance reordering necessary between English and German. Target language reordering, on the other hand, does not

[17] The main clause is the same for all translations and left out for space reasons.

| | |
|---|---|
| source | I endorse the rapporteur's observations that [...] |
| reference | Ich *schließe* mich dem Wunsch des Berichterstatters nach [...] *an*. |
| baseline | Ich *schließe* mich den Auffassungen des Berichterstatters, [...] |
| verb p.; step 1 | Ich *schließe an* mich den Auffassungen des Berichterstatters, [...] |
| verb p.; final | Ich *schließe* mich *an* den Auffassungen des Berichterstatters, [...] |
| source | The Commission proposal also makes provision for including [...] |
| reference | Der Vorschlag der Kommission *sieht* darüber hinaus *vor*, [...] |
| baseline | Der Vorschlag der Kommission *sieht* zudem *vor*, dass [...] |
| verb p.; step 1 | Der Vorschlag der Kommission *sieht vor* zudem, dass [...] |
| verb p.; final | Der Vorschlag der Kommission *sieht* zudem *vor*, dass [...] |

Table 5.19: Sample translations for system with target language reordering of verbal particles. Main verb and verbal particles emphasised.

overcome the problems inherent in SMT, but only shifts them to the second translation step.

What is left to discuss is the system in which only verbal particles have been reordered. Its results are not significantly different from those of the baseline system. Looking at the translations in table 5.19, we find that the problems are the same as in the fully reordered models. The second translation step only successfully restores word order over short distances, generally in cases where the baseline approach is also successful. Additionally, verbal particles typically occur immediately after the main verb. Verbal particles that are at a greater distance from the main verb are rare: we observe this phenomenon in only 2% of all sentences in Europarl.[18] Consequently, the quantitative effect of any measures specifically trying to improve the translation of verbal particles is low. While we still prefer the verbal particle to be in the wrong place instead of missing, the positive effect is too small to warrant the time invested. We do not rule out that target language reordering is profitable in some situations, for instance for language pairs for which source language reordering is not possible.

[18] In total, verbal particles occur in 8% of all sentences.

# 6 Conclusion

## 6.1 Contributions

Investigating SMT from English to German, we explored the strengths and weaknesses of current phrase-based systems. Phrase-based SMT adequately considers local context and contiguous phrases that are best translated as one unit, but fails when word choice depends on long-distance words in the source text or syntactic roles, or when syntactic phrases that constitute a unit of meaning are discontiguous. We proposed several ways of overcoming these shortcomings, with varied success.

A syntactically enriched target text, combined with a language model on syntactic relations, failed to yield any improvement. Since we found the Pro3Gres output ill-suited for this purpose, and the approach has already been investigated in (Koehn and Hoang 2007), we did not pursue it further.

We investigated target language reordering as a possible alternative to source language reordering, and found the latter to be preferable. The two-step approach proved to suffer from the same limitations as the one-step translation did. A correct word order could not be restored since this would have involved aligning contiguous phrases with discontiguous ones, which the Moses system is unable to. We also considered target language reordering where source language reordering is not possible, as is the case for German verbal particles. The positive effect was too small to be relevant in practice (or even statistically significant), and the fact that the second translation step is deficient also holds true in this case.

Syntactic enrichment of the source text produces better translation results in some cases, but suffers from an increase in data sparseness. In consequence, we found a

purely log-linear integration of a factored TM to be ineffective in our experiments. Related research on using syntax in factored, phrase-based SMT confirms the problems. Birch, Osborne and Koehn (2007) conclude that "[using a log-linear model] makes finding good weights difficult as the influence of the general model is greater, and its difficult for the more specific model to discover good weights". Avramidis and Koehn (2008) report that they "tested several various combinations of [input factors]", and that "some combinations seem to be affected by sparse data problems". A close investigation of the log-linear combination of TMs, the identification of reasons why it may fail to obtain an improvement over the baseline, and suggestions to overcome these problems, constitute the main contributions of this thesis.

Firstly, our factored model yielded better results after we filtered out phrase pairs with $count(\bar{s}) < 10$, for which MLE does not estimate good probabilities (mostly due to the noise introduced by word alignment and parsing). Secondly, we saw that the phrase translation probability $\phi(\bar{s}|\bar{t})$ and the lexical weight $lex(\bar{s}|\bar{t})$ may be several orders of magnitude lower in the factored model than in the surface form model, which we consider to be probematic for balancing the two models. Always using the values from the surface form model for these two probabilities did not significantly improve performance when a factored model was used on its own, but is beneficial when working with alternative paths. Foremost, it allowed us to overcome the third problem we identified. MERT overfitted model parameters in experiments with several translation models, and we could not obtain good parameters. Instead, we decided to use baseline parameters for all translation models, and successfully demonstrated an improvement over the baseline scores. Lastly, we showed that a large number of translation models can be combined by implementing a deterministic back-off system, and that performance exceeds that of using only two translation models. Our best system, combining all four measures, obtained a significant improvement over baseline system scores (about 0.35 BLEU points). The improvement is not large by any means, but our experiments highlight that, even if enriching TMs with more data fails to yield an immediate improvement, this does not entail that the data is useless for translating. Rather, the detrimental

effects of data sparseness may outweigh and thus mask the positive potential of data-rich models.

## 6.2 Outlook

We have shown that our heuristics to integrate factored translation models yielded better translation results in one specific setting: a phrase-based SMT system translating from English to German, trained on 20 million tokens of European Parliament proceedings, and enriched with syntactic information obtained by the Pro3Gres parser. There are numerous other scenarios that are worth investigating.

English–German translations are notoriously difficult because of differences in word order and the morphological richness of German. The problems we tried to address with syntactic enrichment are dwarfed by errors introduced during word and phrase alignment. We expect syntactic enrichment to be beneficial even when, and especially when, word alignment is good, since our motivation for syntactic enrichment holds true even if all other models are noise-free. Therefore, adapting the approach to other language pairs might prove profitable.

It is unclear what effect the choice of training corpus, both in terms of size and genre, has on the suitability of factored models. We know that SMT performance heavily depends on training set size, but levels out at some point, so that a further increase in size leads to marginal improvements. Since data sparseness has proven more serious for syntactically enriched models, we expect an increase in training set size to be beneficial even if an surface form system no longer profits from it. If, on the other hand, smaller, task-specific corpora are used for SMT systems, approaches that mitigate data sparseness will have more relevance than syntactic enrichment. Potential users of SMT have good reason to favour a genre-specific corpus, even if it is significantly smaller than Europarl. For instance, a system trained on Europarl is unable to translate casual conversation, and will even fail to translate simple sentences such as *Wie heißt du?*.

On a related note, we expect syntactic disambiguation to become more useful if the training corpus is heterogeneous. Some word forms that are theoretically ambiguous are used exclusively in one sense in Europarl, which reduces the need for word form disambiguation. The most extreme example is *floor*, which is consistently translated as *Wort* in Europarl, while literal translations are so rare that they only take up 2-3% of the probability space.[1] This strengthens our prediction that SMT systems based on large, general-purpose corpora can profit more from syntactic enrichment than those based on small, task-specific ones.

We claimed that parser selection, its underlying grammar and parsing errors have a considerable effect on the performance of the factored models. However, we did not contribute any hard numbers to back up this claim. While we cannot investigate the effectiveness of our approach given perfect parsing, recreating the experiment without the modifications to the parser output we suggested, with artificial noise added to parsing, or with a different parser altogether, would help to illuminate the role of the parser. More generally, having proposed improvements to systems using factored models, we can investigate the inclusion of other linguistic features such as POS tags.

We avoided Minimum Error Rate Training for the systems combining up to four translation models by using the same model parameters for all models. While this did lead to a small improvement over baseline scores, we are confident that these model parameters are not optimal, and that there is further room for improvement. This would require more efficient and reliable ways to optimize a large number of model paramaters, considerably more time, and/or more computational capacity, than at our disposal.

We have proposed filtering out phrase pairs with $count(\bar{s}) < 10$ (fewer than 10 observations of the source phrase during training), from all TMs except for the most general one. We have demonstrated that this yields an improvement in our experiments, but it is unclear in how far these findings can be transferred to other language pairs, corpus sizes, and factors.

[1] This surprising alignment is caused by the dominance of the idiom *take the floor*, translated as *das Wort ergreifen*, or variants thereof.

On a methodological level, we found the evaluation of translation quality in SMT far from reliable. The variability caused by different model parameters (obtained through MERT) proved to be far greater than that of the effects we wanted to measure. Moreover, by reaching a sizable improvement of 0.7 BLEU points by simply normalizing the German complementizer *dass*/*daß*, we can add one more example to the list of measures that improve evaluation scores, but not actual translation quality.

To recapitulate, we were able to show that a syntactic analysis of the source text can improve translation quality. However, the difficulties we encountered during our experiments, particularly the question how to best combine several translation models, forced us to compromise and abandon the aim of fully optimizing the system. This means that we have not yet fully explored the potential of syntactic enrichment, and that further reasearch is likely to yield better results.

# Bibliography


Arun, A. (2007). *Discriminative Training for Phrase-Based Machine Translation*. In: First Machine Translation Marathon. Edinburgh.

Avramidis, E. and Koehn, P. (2008). *Enriching Morphologically Poor Languages for Statistical Machine Translation*. In: Proceedings of ACL-08: HLT. Columbus, Ohio: Association for Computational Linguistics, 763–770.

Banerjee, S. and Lavie, A. (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. Ann Arbor, Michigan: Association for Computational Linguistics, 65–72.

Bertoldi, N., Haddow, B. and Fouet, J.-B. (2009). *Improved Minimum Error Rate Training in Moses*. In: Proceedings of 3rd MT Marathon. Prague, Czech Republic.

Birch, A., Osborne, M. and Koehn, P. (2007). *CCG Supertags in Factored Statistical Machine Translation*. In: Proceedings of the Second Workshop on Statistical Machine Translation. Prague, Czech Republic: Association for Computational Linguistics, 9–16.

Bod, R. (2007). *Unsupervised Syntax-Based Machine Translation: The Contribution of Discontiguous Phrases*. In: Proceedings MT Summit 2007. Copenhagen, 51–57.

Brown, P. et al. (1990). *A Statistical Approach to Machine Translation*. In: Computational Linguistics, 16, Nr. 2, 79–85.

Brown, P. et al. (1993). *The Mathematics of Statistical Machine Translation: Parameter Estimation*. In: Computational Linguistics, 19, Nr. 2, 263–311.

Brown, P., Lai, J. and Mercer, R. (1991). *Aligning Sentences in Parallel Corpora*. In: Proceedings of the 29th Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA, 169–176.

Callison-Burch, C., Osborne, M. and Koehn, P. (2006). *Re-evaluating the Role of Bleu in Machine Translation Research*. In: Proceedings the Eleventh Conference of the European Chapter of the Association for Computational Linguistics. Trento, Italy, 249–256.

Callison-Burch, C. et al. (2008). *Further Meta-Evaluation of Machine Translation*. In: Proceedings of the Third Workshop on Statistical Machine Translation. Columbus, Ohio: Association for Computational Linguistics, 70–106.

Carl, M. et al. (2000). *Towards a Dynamic Linkage of Example-based and Rule-based Machine Translation*. In: Machine Translation, 15, Nr. 3, 223–257.

Carpuat, M. and Wu, D. (2005). *Word Sense Disambiguation vs. Statistical Machine Translation*. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 387–394.

Carpuat, M. and Wu, D. (2007). *Improving Statistical Machine Translation Using Word Sense Disambiguation*. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). Prague, Czech Republic: Association for Computational Linguistics, 61–72.

Cer, D., Jurafsky, D. and Manning, C. (2008). *Regularization and Search for Minimum Error Rate Training*. In: Proceedings of the Third Workshop on Statistical Machine Translation. Columbus, Ohio: Association for Computational Linguistics, 26–34.

Collins, M., Koehn, P. and Kučerová, I. (2005). *Clause Restructuring for Statistical Machine Translation*. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 531–540.

Curran, J. R. and Clark, S. (2003). *Investigating GIS and Smoothing for Maximum Entropy Taggers*. In: EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 91–98.

Dalrymple, M. (1999). *Semantics and Syntax in Lexical Functional Grammar*. Cambridge, Mass: MIT Press.

Doddington, G. (2002). *Automatic Evaluation of Machine Translation Quality Using N-Gram Co-Occurrence Statistics*. In: Proceedings of the Second International Conference on Human Language Technology Research. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 138–145.

Estrella, P. (2008). *Evaluating Machine Translation in Context: Metrics and Tools*. Ph. D thesis, University of Geneva.

Federico, M. and Cettolo, M. (2007). *Efficient Handling of N-gram Language Models for Statistical Machine Translation*. In: Proceedings of the Second Workshop on Statistical Machine Translation. Prague, Czech Republic: Association for Computational Linguistics, 88–95.

Foster, G. and Kuhn, R. (2009). *Stabilizing Minimum Error Rate Training*. In: Proceedings of the Fourth Workshop on Statistical Machine Translation. Athens, Greece: Association for Computational Linguistics, 242–249.

Foth, K. A. (2005). *Eine umfassende Contraint-Dependenz-Grammatik des Deutschen*. Hamburg: University of Hamburg.

Fraser, A. and Marcu, D. (2007). *Measuring Word Alignment Quality for Statistical Machine Translation*. In: Comput. Linguist. 33, Nr. 3, 293–303.

Grover, C. and Tobin, R. (2006). *Rule-Based Chunking and Reusability*. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006). Genoa, Italy.

Haapalainen, M. and Majorin, A. (1995). *GERTWOL und Morphologische Disambiguierung für das Deutsche*. In: Proceedings of the 10th Nordic Conference of Computational Linguistics. Helsinki: University of Helsinki, Department of General Linguistics.

Hardmeier, C. (2008). *Using Linguistic Annotations in Statistical Machine Translation of Film Subtitles*. Master's thesis, University of Basel.

Hasan, S., Bender, O. and Ney, H. (2006). *Reranking Translation Hypotheses Using Structural Properties*. In: Proceedings of the EACL-2006 Workshop on Learning Structured Information in Natural Language Applications. Trento, Italy, 41–48.

Holmqvist, M. et al. (2009). *Improving Alignment for SMT by Reordering and Augmenting the Training Corpus*. In: Proceedings of the Fourth Workshop on Statistical Machine Translation. Athens, Greece: Association for Computational Linguistics, 120–124.

Hutchins, J. (1978). *Machine Translation and Machine-Aided Translation*. In: Journal of Documentation, 34, Nr. 2, 119–159.

Hutchins, J. (1999). *The Development and Use of Machine Translation Systems and Computer-Based Translation Tools*. In: International Conference on Machine Translation and Computer Language Information Processing. Beijing, China.

Johansson, R. and Nugues, P. (2007). *Extended Constituent-to-Dependency Conversion for English*. In: Proceedings of NODALIDA 2007. Tartu, Estonia.

Katz, S. M. (1987). *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer*., 400–401.

Kay, M. (1982). *Machine translation*. In: Computational Linguistics, 8, Nr. 2, 74–78.

Koehn, P. (2002). *Europarl: A Multilingual Corpus for Evaluation of Machine Translation*. unpublished draft.

Koehn, P. (2004). *Statistical Significance Tests for Machine Translation Evaluation*. In: Proceedings of EMNLP 2004. Barcelona, Spain.

Koehn, P. (2005). *Europarl: A Parallel Corpus for Statistical Machine Translation*. In: Machine Translation Summit X. Phuket, Thailand, 79–86.

Koehn, P. and Hoang, H. (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). Prague, Czech Republic: Association for Computational Linguistics, 868–876.

Koehn, P. et al. (2007). *Moses: Open Source Toolkit for Statistical Machine Translation*. In: ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. Prague, Czech Republic: Association for Computational Linguistics, 177–180.

Koehn, P. and Knight, K. (2003). *Empirical Methods for Compound Splitting*. In: EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 187–193.

Koehn, P., Och, F. J. and Marcu, D. (2003). *Statistical Phrase-based Translation*. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Morristown, NJ, USA: Association for Computational Linguistics, 48–54.

Macherey, W. et al. (2008). *Lattice-Based Minimum Error Rate Training for Statistical Machine Translation*. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, Hawaii: Association for Computational Linguistics, 725–734.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press.

Minnen, G., Carroll, J. and Pearce, D. (2000). *Robust, Applied Morphological Generation*. In: Proceedings of the 1st International Natural Language Generation Conference (INLG). Mitzpe Ramon, Israel.

Nießen, S. and Ney, H. (2004). *Statistical Machine Translation with Scarce Resources Using Morpho-syntactic Information*. In: Computational Linguistics, 30, Nr. 2, 181–204.

Och, F. J. (2003). *Minimum Error Rate Training in Statistical Machine Translation*. In: ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 160–167.

Och, F. J. and Ney, H. (2001). *Discriminative Training and Maximum Entropy Models for Statistical Machine Translation*. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 295–302.

Och, F. J. and Ney, H. (2003). *A Systematic Comparison of Various Statistical Alignment Models*. In: Computational Linguistics, 29, Nr. 1, 19–51.

Papineni, K. et al. (2001). *BLEU: A Method for Automatic Evaluation of Machine Translation*. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 311–318.

Popović, M. and Ney, H. (2006). *POS-based Word Reorderings for Statistical Machine Translation*. In: International Conference on Language Resources and Evaluation. Genoa, Italy, 1278–1283.

Schiller, A. et al. (1999). *Guidelines für das Tagging deutscher Textcorpora mit STTS*. Stuttgart: Institut fur maschinelle Sprachverarbeitung – Technical report.

Schmid, H. (1994). *Probabilistic Part–of–Speech Tagging Using Decision Trees*. In: Proceedings of the Conference on New Methods in Language Processing. Manchester, UK.

Schneider, G. (2008). *Hybrid Long-Distance Functional Dependency Parsing*. Ph. D thesis, Institute of Computational Linguistics, University of Zurich.

Sennrich, R. et al. (2009). *A New Hybrid Dependency Parser for German*. In: Proceedings of the German Society for Computational Linguistics and Language Technology 2009 (GSCL 2009). Potsdam, Germany.

Stolcke, A. (2002). *SRILM – An Extensible Language Modeling Toolkit*. In: Seventh International Conference on Spoken Language Processing. Denver, CO, USA, 901–904.

Telljohann, H., Hinrichs, E. W. and Kübler, S. (2004). *The TüBa-D/Z Treebank: Annotating German with a Context-Free Backbone*. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation. Lisbon, Portugal.

Tiedemann, J. (2005). *Optimization of Word Alignment Clues*. In: Natural Language Engineering, 11, Nr. 3, 279–293.

Tinsley, J., Hearne, M. and Way, A. (2007). *Exploiting Parallel Treebanks to Improve Phrase-Based Statistical Machine Translation*. In: Proceedings of Treebanks and Linguistic Theories (TLT '07). Bergen, Norway.

Vogel, S., Ney, H. and Tillmann, C. (1996). *HMM-based Word Alignment in Statistical Translation*. In: Proceedings of the 16th Conference on Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 836–841.

Weaver, W. (1949/1955). *Translation*. In: Locke, W. N. and Boothe, A. D., editors, Machine Translation of Languages. Cambridge, MA: MIT Press, 15–23, Reprinted from a memorandum written by Weaver in 1949.

White, J. S., O'Connell, T. and O'Mara, F. (1994). *The ARPA MT Evaluation Methodologies: Evolution, Lessons, and Future Approaches*. In: Proceedings of the First Conference of the Association for Machine Translation in the Americas. Columbia, MD, USA, 193–205.

Yamada, K. and Knight, K. (2001). *A Syntax-Based Statistical Translation Model*. In: ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 523–530.

# I Sample Translations

| | |
|---|---|
| source | I would thank both of them for their magnificent efforts. |
| reference | Ich danke beiden für ihren großartigen Einsatz. |
| baseline | Ich danke auch für ihre enormen Anstrengungen. |
| experiment | Ich danke sowohl von ihnen für ihre hervorragende Arbeit. |
| source | I think the suggestion is worth looking into. |
| reference | Ich halte den Vorschlag für diskussionswürdig. |
| baseline | Ich denke, der Vorschlag geprüft wird. |
| experiment | Ich denke, der Vorschlag geprüft. |
| source | You have addressed the important question of contract compliance. |
| reference | Sie haben die wichtige Frage der Vertragseinhaltung aufgegriffen. |
| baseline | Sie haben die wichtige Frage der Vertrag. |
| experiment | Sie haben die wichtige Frage des Vertrags. |
| source | However the EPLP has strong reservations about the following areas. |
| reference | Starke Vorbehalte hat die EPLP jedoch zu den folgenden Bereichen: |
| baseline | Die EPLP hat folgenden Bereichen starke Vorbehalte. |
| experiment | Die EPLP hat jedoch den folgenden Bereichen starke Vorbehalte. |
| source | What are the basic strands which make up the content? |
| reference | Welches sind nun die wichtigsten inhaltlichen Merkmale? |
| baseline | Was sind die grundlegenden Elementen, die den Inhalt? |
| experiment | Was sind die wichtigsten Bereiche, die den Inhalt? |
| source | (The sitting was closed at 10.56 a.m.) |
| reference | (Die Sitzung wird um 10.56 Uhr geschlossen.) |
| baseline | (Die Sitzung wird um 10.56 Uhr wieder aufgenommen.) |
| experiment | (Die Sitzung wird um 10.56 Uhr geschlossen.) |

Table I.1: Five translations differing between baseline and best system with factored input and deterministic back-off (see table 5.12). Random selection among those short enough to fit on one line.

| | |
|---|---|
| source | In fact, these appropriations are urgently needed! |
| reference | Dabei sind diese Mittel dringend notwendig! |
| baseline | Diese Mittel sind in der Tat dringend erforderlich! |
| experiment | In der Tat, diese Mittel sind dringend erforderlich! |
| source | At present, 1.2 million lorries cross the Brenner Pass every year. |
| reference | Es rollen heute jährlich 1,2 Millionen Lkw über den Brenner-Paß. |
| baseline | Über 1,2 Millionen Lkws über den Brenner pro Jahr. |
| experiment | Derzeit 1,2 Millionen Lkws über den Brenner pro Jahr. |
| source | Is that in any way related to aviation? |
| reference | Betrifft sie den Luftverkehr? |
| baseline | Das ist in keiner Weise im Luftverkehr? |
| experiment | Ist das in keiner Weise im Luftverkehr? |
| source | I think that you could allow me that much time. |
| reference | Die können Sie mir zugestehen, glaube ich. |
| baseline | Ich glaube, sie könnten, sehr viel Zeit. |
| experiment | Ich glaube, sie könnten mir sehr viel Zeit. |
| source | He proposes increasing the budget funding. |
| reference | Er schlägt mehr Haushaltsmittel vor. |
| baseline | Er schlägt vor, den Haushalt zu finanzieren. |
| experiment | Er schlägt eine Ausweitung des Haushalts finanziert werden. |

Table I.2: Five translations differing between baseline and system with factored output (see table 5.15). Random selection among those short enough to fit on one line.