Karlsruhe Institute of Technology

Extending Phrase-Based Machine Translation with Topic Models

Diploma Thesis of

Isabel Slawik

At the Department of Informatics
Interactive Systems Lab

Reviewer:	Prof. Dr. Alexander Waibel
Second reviewer:	Dr. Sebastian Stüker
Advisor:	Dipl.-Inform. Jan Niehues

Duration: 01 July 2012 – 31 December 2012

Selbstständigkeitserklärung

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten Anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, 19.12.2012

....

(Isabel Slawik)

Zusammenfassung

Die Disambiguierung von Homonymen stellt ein schwieriges Problem für maschinelle Übersetzungssysteme dar. Homonyme sind Wörter, die, je nach dem Kontext in dem sie benutzt werden, unterschiedliche Bedeutungen haben. Häufig werden Homonyme entsprechend ihrer Bedeutung in andere Wörter der Zielsprache übersetzt. Stand der Technik sind allerdings immernoch Übersetzungssysteme, die jeden Satz für sich ohne weiteres Kontextwissen übersetzen.

Ziel dieser Arbeit ist die Disambiguierung von Homonymen mithilfe von themenbasierten Modellen in einem phrasenbasierten maschinellen Übersetzungssystem. Der verwendete Ansatz kombiniert dabei erstmals *Topic*-Modelle mit einem *Discriminative Word Lexicon* (DWL). In einem DWL wird für jedes Zielwort ein *Maximum Entropy*-Klassifikator trainiert, der entscheidet, ob das Zielwort in der Übersetzung eines gegeben Quellsatzes vorkommen sollte. Dabei dienen die Worte des Quellsatzes als Merkmale für die Klassifizierung. In diesem Ansatz wird jedem Quellsatz ein spezielles Wort für das wahrscheinlichste Thema hinzugefügt. Das DWL berücksichtigt dadurch das Thema, welches durch denn Kontext, in dem der Satz steht, erkannt wurde, bei der Entscheidung.

Zuerst wird ein monolinguales Topic-Modell auf den Quelldaten trainiert. Das wahrscheinlichste Thema wird in jeden Quellsatz des dazugehörigen Dokuments eingefügt. Auf den annotierten Quelldaten wird dann ein DWL trainiert. Dabei werden die Themen vom DWL als normales Quellwort behandelt. Die annotierten Testdaten werden anschließend von einem Übersetzungssystem mit dem erweiterten DWL übersetzt.

Als Datenbasis für das System wurden Transkripte von TED-Konferenzvorträgen benutzt. Die Übersetzungen von Deutsch nach Englisch wurden mit der BLEU-Metrik bewertet. Einige Konfigurationen erreichten eine leichte Verbesserung um 0.1 BLEU-Punkte gegenüber dem Vergleichssystem. Neben einer ausführlichen Studie der Parameter des Topic-Modells und ihres Einflusses auf die Übersetzungsleistung wurden spezielle Konfigurationen ausgehend von alternativen Merkmalen getestet. Die Unzulänglichkeiten der BLEU-Metrik zur Bewertung der Disambiguierungsaufgabe werden besprochen und zwei alternative automatische Evaluationsmetriken vorgestellt. Da eine Evaluation des Systems durch Menschen im Rahmen der Arbeit aufgrund des damit verbundenen Aufwands nicht durchführbar war, werden die Vor- und Nachteile des Systems anhand von ausgewählten, charakteristischen Übersetzungsbeispielen dargelegt. Zusätzlich zu den TED-Daten wurde eine Evaluation auf Daten des Quaero-Projekts für die Übersetzungsrichtung von Deutsch nach Französisch durchgeführt.

Abstract

Homonyms are a challenging problem for statistical machine translation systems because they are translated into different words depending on their meaning. Current machine translation systems operate with no notion of context beyond the sentence-level. But without context it is often impossible to decide the correct translation of a homonym. We provide this context through the use of a topic model trained with Latent Dirichlet Allocation.

In this work we investigated ways to incorporate topics into a phrase-based machine translation system to aid in the disambiguation of homonyms. Our novel approach is based upon the idea of combining a topic model with a Discriminative Word Lexicon (DWL). A Discriminative Word Lexicon is a maximum entropy classifier predicting target words based on the words of the source sentence. It suffices to train a monolingual topic model in our approach. An additional advantage is the easy inclusion of our approach into existing log-linear model frameworks. Our model avoids data sparsity concerns by exploiting all the training data available and is highly scalable for multiple target languages.

We first train a monolingual topic model on the source side of our parallel documents. The most likely topic for every document is then added to the source text through the use of special topic markers. By including topic markers denoting the most likely topic in our documents, we utilize context information in the decision of the correct target words. A Discriminative Word Lexicon is then trained on this topic-augmented source text. This results in the inclusion of topics as source features in the DWL.

We report experiment results on a German to English translation task using the TED talks corpus. Slight improvements in BLEU score of up to 0.1 BLEU points can be observed in a simultaneous translation scenario. In addition to an extensive parameter study to find an optimal topic model configuration, we present several specific configurations trained on different DWL feature sets. We discuss the shortcomings of the BLEU score to measure the performance of a translation system regarding the disambiguation of homonyms and consider two additional automatic evaluation measures as alternative information sources. In lieu of a full human evaluation, the merits and deficits of our system are highlighted using hand-picked characteristic example translations. Furthermore we report experiment results on a large-scale German to French translation task.

Contents

1 Introduction 1
1.1 Motivation 1
1.2 Task of Word Sense Disambiguation 2
1.3 Goal of this Work 3
1.4 Overview 4

2 Fundamentals 7
2.1 Machine Translation 7
2.1.1 Statistical Machine Translation 7
2.1.2 Phrase-Based Machine Translation 8
2.2 Discriminative Word Lexicon (DWL) 9
2.3 Topic Models 10
2.3.1 Latent Dirichlet Allocation (LDA) 11

3 Related Work 15
3.1 Our Work 16

4 Approach 19
4.1 Incorporation of Topic Models 19
4.1.1 Training a Topic Model 19
4.1.2 Training a DWL 21
4.1.3 Running a New Configuration 22
4.2 Topic Model Parameters 22
4.2.1 Number of Topics 23
4.2.2 Number of Topic Markers 23
4.2.3 Including Probability 23
4.2.4 Stop Lists 24
4.2.5 Segmentation 24
4.2.6 MERT Weights 27

5 Evaluation 29
5.1 Data 29
5.2 Baseline System 31
5.3 Number of Topics 31
5.4 Multiple Topic Markers and Probabilities 33
5.5 Segmentation of Source Texts 34
5.6 Stop Lists 36
5.7 MERT Weights 39
5.8 Special Configurations 40
5.8.1 Keywords 40
5.8.2 Aligned Words 42
5.8.3 Only Topic Markers 43

5.9 Additional Evaluation Metrics 44
5.9.1 Word Accuracy 44
5.9.2 Oracle Scores 46
5.10 Translation Examples 48
5.11 Poisoned Cookie 49
5.12 German to French Translation Task 51

6 Conclusion **53**
6.1 Summary 53
6.2 Outlook 54

Bibliography **57**

1. Introduction

1.1 Motivation

The human need to communicate is ingrained in each and every one of us. Communication is often difficult enough when we talk to someone in our own language. Whenever we meet someone who speaks a different language than us, our ability to communicate is suddenly limited to what we can express with gestures, grimaces and sounds. In order to share ideas that go beyond basic concepts, translation is needed. This is often done by finding a person who can speak both languages and asking them to interpret. However, in an increasingly globalized world, the need for translation has vastly exceeded the number of available translators.

Consider the European Union, which has 23 official languages. This means that every piece of legislation as well as all speeches given in the European Parliament have to be translated into every one of these languages. With the addition of new member states to the EU, the number of official languages keeps growing. The resulting need for translation is huge and costly. Another huge market for translation is today's global economy. Multinational corporations operate in many different countries, requiring communication not only with local suppliers, customers and business partners, but also between different branches and teams of the company itself, often spanning many languages. Even many smaller companies often deal with foreign markets, importing supplies from all over the world and exporting their goods to consumers abroad. Nowadays you are only one mouse click away from ordering electronic goods from China, jewelry from India or the newest cellphone from the USA. Aside from enabling you to conveniently shop at home, the Internet itself contains a huge well of information from all over the world. However, to access the information you need to understand the language it is written in. Other markets with obvious applications for translation services include tourism, disaster relief and humanitarian aid.

Machine translation provides a way to translate texts whenever the use of a human translator would be too costly, slow, cumbersome or one plainly cannot be found. Unfortunately, devices like the telepathic force field of the TARDIS in Doctor Who, the babelfish in Douglas Adam's Hitchhiker's Guide to the Galaxy or the Universal Translators used in Star Trek which can produce flawless translations between any language pair imaginable still remain science fiction to this day. While machine translation has made considerable progress in the last decades, we still have a long way to go in creating mobile devices that can automatically produce fluent, semantically correct translations between every language pair you might be needing in that moment.

Statistical machine translation is a very promising approach to machine translation, but it still has its limits. Current statistical machine translation systems are built for one specific translation direction and require large parallel corpora as training data. While they are able to produce sentences that may not always be grammatically correct, the translation quality will often suffice to get the point across and facilitate a rudimentary understanding of the source text. However, there are still numerous instances where the meaning of the original sentence is lost in translation.

1.2 Task of Word Sense Disambiguation

One particular difficulty in machine translation lies in the disambiguation of homonyms. Homonyms are words that are spelled the same, but have different meanings. The English word *shot* in the sentence *That was a nice shot!* can refer to a bullet fired with a gun or rifle, a picture taken with a photo camera, or the kick, throw or batting of a ball in soccer and similar sports.

Sometimes the translation for a homonym is also a homonym with the same meanings in the target language. The German word *Schuss* can be used for all three meanings of the word *shot* and we could therefore use it in a translation without worrying about the exact meaning of *shot*. However, most of the time homonyms are translated into different words in the target language according to their meaning.

Current machine translation systems translate one sentence at a time. This means that we translate devoid of any context of the document in which our sentence appeared or the content of the surrounding sentences. The problem becomes evident in a small example. Consider the following sentence:

She's got class.

Given just this sentence, we cannot decide its meaning because we are lacking context. *To have class* could either refer to a lecture in our unnamed heroine's schedule or to our heroine having a certain elegance about her. Depending on the sense of *class*, we would have to choose a different word in German to retain the original meaning of the sentence. In a simple machine translation system we would have no way of deciding which translation would be the correct one, leaving us with no choice but to guess. This means that we would lose the meaning of the original sentence in many of the translations we produce. What is worse, is that we would produce an entirely correct sentence, so the human user of our system would have no way of knowing that we actually made a mistake. To see how we can correctly dissolve this ambiguity, let us have a look at the context of our sentence.

The previous sentence might be something along the following lines:

Mary doesn't have time for the study group on Tuesdays. She's got class.

From this sentence, we learn that Mary does not have an opening in her schedule for study group, and and the reason that was stated in our original sentence is that she has class. With this information, we now know that *class* refers to a lecture and we could therefore use the German word *Unterricht* as the correct translation of *class*.

Let us consider another preceding sentence that might tell a different story:

Mary always looks beautiful. She's got class.

In this case, we can take *class* to mean Mary's elegance, because the previous sentence talks about Mary's appearance. Knowing this, the correct translation for *class* would be the German word *Stil*.

We have seen that analyzing the previous sentence can help us determine the meaning of a homonym. We might now be inclined to always consider the previous sentence when translating. But what if the helpful clue was mentioned in the following line or even a couple of lines away from our original sentence? If we had known that the sentence was from a text about college, we would have been able to choose *Unterricht* as the more likely translation. Similarly, if we knew the sentence was from a text on fashion, we could have picked *Stil* as the correct translation.

Topic models are statistical models that analyze the words in a text to infer the underlying subject of the text, such as *fashion* or *college*. This knowledge can help us in the disambiguation of homonyms by providing context even though the vital clues might be more than a line removed from our sentence.

1.3 Goal of this Work

The goal of this work is to investigate ways of incorporating topic models in phrase-based statistical machine translation using a Discriminative Word Lexicon.

Translation is very seldom a one-to-one correspondence between words in the source and target language. One look in a translation dictionary shows multiple translation options for nearly every word in a language. Often there will be a number of valid translations for each word, where the choice amongst these only influences the mood of a sentence without distorting its meaning. The wrong translation of a homonym, however, can greatly alter or even reverse the meaning of the original sentence, and in the worst case the meaning will be lost altogether.

While these extreme cases only appear few and far between, when they happen they have a grave impact on a human user of the system. In a simultaneous lecture translation, the wrong translation of a homonym can cause quite an embarrassment, whereas in a speech-to-speech translation task it might prohibit communication altogether. While the correct disambiguation of homonyms often does not reflect in automatic evaluation metrics because of their rare occurrence, it is an important task to ensure the usability of our system in a real world application.

The correct choice of words often depends on the context of the word. Most state-of-the-art machine translation systems operate on a sentence level, translating every sentences without any knowledge of the surrounding sentences. Without any further context beyond the single sentence, even a human cannot always tell the correct translation choice, as evidenced in the previous example (cf. Section 1.2). To correctly disambiguate homonyms, we need to incorporate context beyond the single sentence into our machine translation system. Topic models are one way to introduce the needed context into our system.

In this work we try a new approach to topic adaptation in machine translation by incorporating topic models into a Discriminative Word Lexicon. We chose a Discriminative Word Lexicon because its purpose is to predict target words given the set of words in the source sentence. Unlike an n-gram language model, which only has a narrow context of the three to four preceding words, the DWL has a global view of our source sentence.

The main advantages of our approach are as follows:

- The DWL requires only the training of a **monolingual topic model** on the source side of our corpus, unlike language model adaptation approach which necessitate some form of bilingual topic models.

- Our approach seamlessly incorporates into a Discriminative Word Lexicon, which is **easily integrated into existing frameworks** as an additional component in the log-linear model.
- By using all the data available to us, we **avoid data sparsity problems** that can occur with approaches that filter the trained models according to the topic. This is especially important working on small corpora such as the TED corpus that we used in this thesis.
- Another advantage is the **scalability** of our approach. We only need to train one topic model per source language, regardless of the number of target languages. Consider the task of translating a speech from the European Parliament, where every speech has to be translated into over 20 languages. With our approach we only need to train one topic model on the original speech, which can then be used to adapt all 20 translation models.

In this thesis we developed a way to use topic information in a Discriminative Word Lexicon to help disambiguate homonyms. We achieved this by training a topic model on our source text using Latent Dirichlet Allocation. The source data is then augmented with topic markers, denoting the most likely inferred topics for each document. These topic markers are then used as additional source features in a Discriminative Word Lexicon, guiding the word choices of our decoder.

In order to achieve the best translation results possible with our approach, a number of parameters has to be considered in the training of a topic model. We methodically investigate the effects of different parameter values on translation performance using numerous topic models. In addition to this, we review several special configurations beyond the parameter range to investigate specific aspects of our models. We evaluate our approach using automatic evaluation metrics and analyze additional metrics to gain further insight into the performance of our configurations. Lastly, we discuss the shortcomings of automatic evaluation metrics as quality measures for the task of homonym disambiguation and demonstrate typical problems with the disambiguation of homonyms using example translations of our system.

1.4 Overview

The rest of this thesis is structured as follows:

Chapter 2 gives an introduction to the fundamental concepts that form the basis of this work. It starts by describing the process of statistical machine translation in general and phrase-based machine translation in particular. Afterwards the Discriminative Word Lexicon model is explained. The chapter concludes with a presentation of topic models with an emphasis on Latent Dirichlet Allocation.

Chapter 3 gives an overview of previous works in the field of topic adaptation in machine translation and shows how these relate to our approach. A brief history of the field is given before the two prevalent approaches to this problem are presented: language model adaptation and translation model adaptation. Advantages and drawbacks of state-of-the-art examples for both approaches are analyzed. Finally the approach used in this thesis is contrasted with the previously presented works.

Chapter 4 describes how we incorporated topics into the translation process. The training procedures for topic models and Discriminative Word Lexicons are laid out. We also describe the typical process required to run a new configuration. In the last section of the chapter the parameters we studied in our experiments are presented.

Chapter 5 provides the results gained in our experiments. We first describe the data used in the main part of our experiments, the TED corpus, and outline the design of our baseline system. In addition to the discussion of the effects individual parameters have on translation performance, multiple special-case configurations are presented and evaluated. To gain further insight into the performance of the proposed configurations, we present characteristic translation examples and review two additional performance metrics. We also discuss the shortcomings of automatic evaluations. The chapter concludes with the presentation of our results on a large-scale translation task from German to French from the Quaero 2012 evaluation.

Chapter 6 summarizes the work and its most important findings presented in this thesis. This thesis is concluded with an outlook on possible future work using topic adaptation with Discriminative Word Lexicons.

2. Fundamentals

This chapter aims to briefly introduce the basic concepts of statistical and in particular phrase-based machine translation, discriminative word lexicons and topic models using the example of latent Dirichlet allocation to aid the reader in the understanding of this thesis. For an in-depth discussion of the concepts mentioned here, the interested reader is kindly referred to the literature mentioned in the respective sections.

2.1 Machine Translation

Before we introduce phrase-based machine translation (PBMT), we discuss the general process of statistical machine translation (SMT) on which PBMT is based. [Koe10] gives a comprehensive overview and detailed discussion of techniques used for SMT.

2.1.1 Statistical Machine Translation

The problem of machine translation is the task of finding a translation from a sentence in the source language to a sentence in the target language. Statistical machine translation was first proposed by Brown et al. [BPPM93] at IBM. They describe a series of increasingly complex models for word-to-word translation which became later known as the *IBM Models*. In their paper they proposed the fundamental equation of machine translation, called the *noisy-channel model* in analogy to information theory:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e} \in E} p(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e} \in E} p(\mathbf{f}|\mathbf{e}) \cdot p(\mathbf{e}) \tag{2.1}$$

Given a source sentence $\mathbf{f} = f_1, ..., f_J$ in the source language, we want to find the most probable target sentence $\mathbf{e} = e_1, ..., e_I$ in the target language given the set of all target sentences E. Equation 2.1 has split the problem of translation up into three parts: the *language model* $p(\mathbf{e})$, the *translation model* $p(\mathbf{f}|\mathbf{e})$ and the *decoder* that performs the search for the arg max.

The language model is a measure for how likely the target sentence would be in the target language. A good language model score implies a fluent output in the target language. It is often modeled by *n-grams* where the probability of the current word depends on the $n - 1$ preceding words. Since the language model depends only on the target language, it can be learned on monolingual corpora.

The translation model is a measure for how likely the target sentence is a translation of the source sentence. A good translation model score implies an accurate output in the target language. Instead of word-to-word translation probabilities as used by Brown et al., the translation model is nowadays composed of phrase translation probabilities learned from large bilingual corpora.

Lastly the decoder has to find the most probable translation hypothesis for the source sentence given the individual model scores.

Today, the *log-linear model* has largely replaced the noisy-channel model. The general form of a log-linear model is given in equation 2.2.

$$\hat{\mathbf{e}} = \arg\min_{\mathbf{e}\in E} \sum_{i=1}^{n} -\lambda_i h_i(\mathbf{e}) \tag{2.2}$$

The log-linear model is a generalization of equation 2.1. Subsequently, the translation and language models can be defined as feature functions in this log-linear model:

$$h_{TM}(\mathbf{e},\mathbf{f}) = \log p(\mathbf{f}|\mathbf{e}) \tag{2.3}$$
$$h_{LM}(\mathbf{e}) = \log p(\mathbf{e}) \tag{2.4}$$

One main advantage of log-linear models is the easy inclusion of additional components, such as a *reordering model*, *word* or *phrase count penalties* and so on. We assume the feature functions $h_i(\mathbf{e})$ to be independent of each other and thus they can be trained separately, allowing for parallelization. The introduction of weights λ_i allows the tuning of the models, putting more emphasis on more fluent or accurate output as needed. An algorithm optimizing the feature weights according to an automatic error metric called *Minimum Error Rate Training* (MERT) was introduced by [Och03].

2.1.2 Phrase-Based Machine Translation

Translating sentences word by word as proposed by Brown et al. is often problematic, due to a number of reasons, the simplest being that most language pairs do not have a straightforward correspondence of words. By translating phrases instead of words, the translation process is greatly simplified. Phrases may consist of a different number of words on the source and target side, thus eliminating the need to deal with deletions or insertions explicitly in the model. They are also able to implicitly model reordering of words within the phrase and can even capture short idioms. Aside from making the translation process simpler, phrase-based models have been shown to significantly outperform word based models in terms of translation quality. They are used in most state-of-the-art SMT systems.

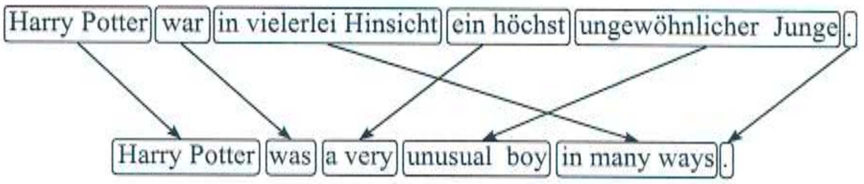


Figure 2.1: An example of phrase-based translation from German to English, showing the alignment and reordering of phrases.

In phrase-based machine translation, the source sentence is split into non-overlapping phrases. They consist of one or more consecutive words and do not have to correspond to meaningful phrases in a linguistic sense. The source phrases are then translated one-to-one to target phrases using a dictionary of possible translations, the phrase-table, and may be reordered. An example how a phrase-based translation might look like is given in Figure 2.1, using the first sentence of the German and English editions of the book "Harry Potter and the Prisoner of Azkaban" by J.K. Rowling.

2.2 Discriminative Word Lexicon (DWL)

A *Discriminative Word Lexicon* (DWL) is a statistical classifier that tries to predict whether to include a target word in the hypothesis given the full source sentence. It was first proposed by Mauser et al. [MHN09]. In this work they trained a *maximum entropy model* for each target word, using one feature per source sentence word. Unlike most translation models used in phrase-based machine translation, the DWL utilizes information across phrase boundaries on a sentence level. This allows it to capture even long-distance dependencies common for example in German, where the prefix of a verb can often be detached from the verb and moved all the way to the end of the sentence.

It is important to note that the DWL does not capture word order, but simply operates on the set of source and target words. Mauser et al. argue that the correct word order of the target sentence is insured by the language model. The commonly used n-gram language models are only able to capture a local context, usually between three to five words. The DWL augments this by using global (on a sentence-level) features.

Mauser et al. model the probability of a target word to be included in the hypothesis as described in equation 2.5.

$$p(e^+|\mathbf{f}) = \frac{\exp\left(\sum\limits_{f\in\mathbf{f}} \lambda_{f,e^+}\phi(f,\mathbf{f})\right)}{\sum\limits_{e\in\{e^+,e^-\}} \exp\left(\sum\limits_{f\in\mathbf{f}} \lambda_{f,e}\phi(f,\mathbf{f})\right)} \tag{2.5}$$

e^+ and e^- are the events that e is or is not included in the target sentence, respectively. The $\lambda_{f,\cdot}$ are the corresponding feature weights. Mauser et al. propose simple binary feature functions as shown in equation 2.6. It is also possible to use weighted feature functions by replacing the binary "1" with the appropriate weight.

$$\phi(f,\mathbf{f}) = \begin{cases} 1 & \text{if } f \in \mathbf{f}, \\ 0 & \text{else} \end{cases} \tag{2.6}$$

Given the probability for a single target word, the probability of a whole sentence can be computed as follows:

$$p(\mathbf{e}|\mathbf{f}) = \prod_{e\in\mathbf{e}} p(e^+|\mathbf{f}) \cdot \prod_{e\in V_e\setminus\mathbf{e}} p(e^-|\mathbf{f}) = \prod_{e\in\mathbf{e}} \frac{p(e^+|\mathbf{f})}{p(e^-|\mathbf{f})} \cdot \prod_{e\in V_e} p(e^-|\mathbf{f}) \tag{2.7}$$

During runtime, it is necessary to score partial translation hypotheses to allow pruning of the search space. In that case it would be impossible to calculate $\prod_{e\in V_e\setminus\mathbf{e}} p(e^-|\mathbf{f})$, since it's not clear which words will eventually be included in the hypothesis. This makes it necessary to use the second transformation, eliminating the term. The second transformation has the added benefit that the product over all the words in the vocabulary is the same for every hypothesis and may therefore be disregarded when scoring the hypotheses.

Since the classifiers of the target words do not depend on the local context, they can be trained concurrently. This is especially important since each word of the target vocabulary is its own class for the classification task. To counter this problem, Mediani et al. [MCN$^+$11] propose a pruning method for the training features. A hypothesis will only contain target words that occur in phrases whose source side matches the source sentence. When training a new classifier for a target word, we can limit ourselves to the sentences that match at least one phrase in which the given target word occurs. In addition to speeding up the training process, Mediani et al. have also found it to improve translation quality.

2.3 Topic Models

Topic models are generative models which work on the assumption that a document consists of a mixture of inherent topics. In this work we use latent Dirichlet allocation to model our topics. [Ble12] is an easy to understand introduction into topic models and latent Dirichlet allocation.

Intuitively speaking, a *topic* is a set of words that often appear in close context with each other. In the mathematical context of our model, a topic is a probability distribution over the words in our *vocabulary*. This means that for each topic, every word in our vocabulary has a different probability of cropping up. For example, we would expect the probability for the word "baseball" to occur in a topic centered around sports to be significantly higher than its probability in a topic about farming. When we say a topic is centered around sports, it means that the topic assigns a high probability to words we would relate with sports. While statistically found topics will often correspond to recognizable, sensible topics for a human, it does not necessarily have to be the case.

Table 2.1 shows the 15 most probable words of six hand-selected topics of 100 topics generated from the TED corpus (cf. Section 5.1). It is easy to assign a "label" to the first four topics, which show words clearly clustered around a human concept such as "belief" or "astronomy". However, it is not quite as easy to find a label for the fifth topic, marked TOPIC_96. It shows a few words one might relate to problem-solving, such as "problem", "wrong", and "thought", but it also has seemingly random words like "sandwich" and "cheese" with a high probability. There is no discernible cohesion to the words in this topic by looking at these words alone. The method to this madness becomes apparent when we look at our corpus, which contains a talk with a high probability for TOPIC_96. It's a talk by Rebecca Saxe entitled "How we read each other's minds"[1] and in this talk she makes her point using a story about two pirates and a cheese sandwich.

Table 2.1 also shows that one word might have a high probability in more than one topic. While "life" pertains to both belief and genetics, the word "space" actually gains a different meaning depending on the topic it occurs in. In an article about astronomy and physics, "space" would most likely be correctly translated into the German "Weltall", meaning "outer space". If it is used in an article about housing and development, a more probable translation would be "Raum", meaning "room". Topic models can therefore be used to guide us in choosing the correct words and translations.

[1] `http://www.ted.com/talks/rebecca_saxe_how_brains_make_moral_judgments.html`

2.3.1 Latent Dirichlet Allocation (LDA)

Latent Dirichlet allocation (LDA) was first introduced by [BNJ03], where they used it for document modeling and text classification. While LDA is often used on text corpora, it can actually work on numerous kinds of discrete data and has been used for image clustering and analysis of genetic data.

LDA assumes that every *document* consists of multiple (latent) topics in different proportions and can therefore be defined as a mixture over topics. While all documents draw from the same fixed pool of topics, the mixture weights will differ from each other. An article about sailing might have a high probability for the topics "ocean" and "weather", whereas a talk about deep-sea wildlife might have most of its probability mass concentrated on the topics "biology", "ocean" and "research".

In order to take a closer look at LDA, we first need to define some terms.

corpus

a collection of M documents $\mathcal{D} = \{\mathbf{w}_1, ..., \mathbf{w}_M\}$

document

a sequence of N words $\mathbf{w} = (w_1, ..., w_N)$, with w_n being the nth word

word

the smallest unit of discrete data indexed by its position in the vocabulary, with w^i being the ith word in a vocabulary of size V

LDA is most easily defined by its generative process, the probabilistic procedure by which the model assumes a new document was created. Algorithm 2.3.1 shows the basic model of LDA according to [BNJ03].

"belief"	"military"	"housing"	"genetics"	"astronomy"	TOPIC_96
god	war	building	dna	universe	people
bible	evil	materials	genes	space	consciousness
religion	force	material	genetic	particles	metaphor
religious	military	space	information	dimensions	problem
faith	power	house	change	theory	wrong
life	iraq	natural	truth	physics	moral
book	soldiers	built	sort	charge	sandwich
hand	study	process	machines	pattern	brain
purpose	kill	block	sequence	particle	cheese
church	inside	street	cell	force	minds
word	american	architecture	understand	shape	show
influence	situation	environment	molecule	beautiful	detail
bread	battle	end	life	numbers	change
darwin	peace	buildings	structure	extra	taking
atheist	guy	inside	scientific	gravity	thought

Table 2.1: Six examples of topics generated from the TED corpus, showing the 15 most probable words for each topic.

```
For each of the $M$ documents $\mathbf{w}_m$ in a corpus $\mathcal{D}$:
1. Choose $N \sim \text{Poisson}(\xi)$
2. Choose $\theta \sim \text{Dirichlet}(\alpha)$
3. For each of the $N$ words $w_n$:
    a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$
    b) Choose a word $w_n$ from $p(w_n|z_n, \beta)$
```

Algorithm 2.3.1: The generative algorithm of Latent Dirichlet Allocation.

For each document, the length (= number of words) of this document is drawn from a suited probability distribution. The type of probability used has no great bearing on the rest of the algorithm. Afterwards, the topic mixture θ is drawn from a Dirichlet prior with the parameter α. The number of topics k and therefore the dimensionality of the Dirichlet distribution is assumed to be known and fixed. To generate the words in a document, we first pick a topic z_n for each word from the topic mixture, and then draw a word w_n from the chosen topic. β is a $k \times V$ matrix where every row contains a topic distribution $\beta_{i,j} = p(w^j|z^i)$.

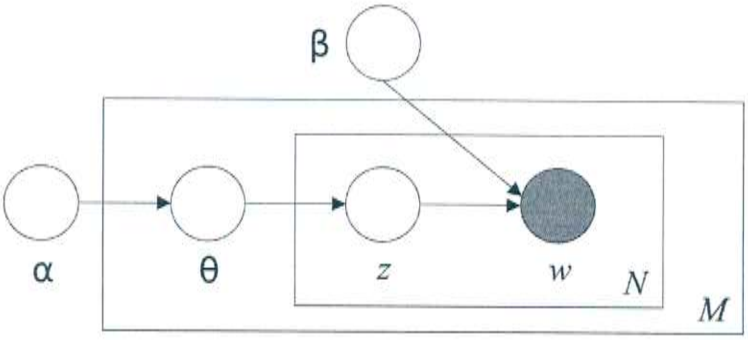


Figure 2.2: A graphical model of LDA using plate notation, taken from [BNJ03].

Figure 2.2 shows a graphical representation of LDA using plate notation. The nodes stand for random variables while the edges model the dependencies between the variables. Shaded nodes represent observed variables, in this case only the words w, while unshaded nodes stand for latent variables. The boxes, called plates, indicate replication of the contained subgraph with the number of repetitions written in the corner. This allows a compact representation of the repeated variables. α and β are sampled once on a corpus-level. The outer plate is repeated M times, once for each document, while the topic and word variables of the inner plate are replicated N times, once for each word in a particular document.

It is important to note a few assumptions of LDA.

1. The occurrences of topics are assumed to be uncorrelated. This is a simplification that does not hold true in real life. We would expect a document that is about "war" to have a higher probability to be also about "politics" than to be about "cooking".
2. The order of words in the documents is neglected. This is called the "bag-of-words" assumption. For our purposes it only matters which words occur, but not where they

occur in the document. LDA also neglects the order of documents in the corpus.

3. The number of topics is presumed to be known and fixed beforehand. This shortcoming can be overcome by using a hierarchical Dirichlet process [TJBB06], which automatically determines the number of topics during inference.

The generative process of LDA describes how we could build a corpus of documents, given a topic structure. In reality, we want the opposite: we already have a corpus of documents and would like to infer the most probable topic structure that might have created this corpus. This means we have to reverse the generative process. However, computing the posterior of LDA is intractable and therefore not possible. We can only approximate it with sampling-based or variational methods.

3. Related Work

There have been a number of approaches to include topic adaptation in machine translation in the past. In this chapter we will analyze a selection of works that have dealt with topic adaptation and try to give an overview of the current state of the field. After reviewing past approaches, we will contrast our work against the previous works and highlight the main concepts of our approach.

There are many ways to apply topic models to machine translation, most of which follow one of two main concepts: language model adaptation or translation model adaptation. Language model adaptation has been inspired by the field of automatic speech recognition and uses topic-dependent n-gram probabilities to adapt a larger background language model. Translation model adaptation, on the other hand, focuses on adapting translation-specific concepts such as the phrase table or translation lexicons.

Language model adaptation has been successfully used in automatic speech recognition for over a decade. In 1999, Gildea and Hofmann [GH99] used Probabilistic Latent Semantic Analysis (PLSA) to construct a unigram topic model which they then combined with a standard trigram language model. Federico [Fed02] extended this approach by combining PLSA with Minimum Discrimination Information (MDI) estimation. Hsu and Glass [HG06] used a Hidden Markov Model (HMM) with LDA to construct topic-dependent trigram language models. To account for changing topics over time, they dynamically changed the mixture weights of their linearly interpolated language models over time using exponential decay.

Unlike automatic speech recognition, adaptation of the language model in machine translation is not straightforward because of the language disparity: The incoming data used to infer adaptation parameters is in the source language, while the language model which is to be adapted is in the target language. Ruiz and Federico [RF11] extend Federico's work [Fed02] to machine translation. They use cross-language topic adaptation to adapt a background language model via MDI estimation. They construct documents from the bilingual corpus containing parallel sentence pairs and train a PLSA topic model on these documents. In testing, the source language text can be used to infer a word-document distribution from which all the source language words are removed. From the remaining target language words a unigram language model is constructed and used to adapt the background language model.

Bilingual models use source-side information to adapt the target language models in one decoder pass. Tam et al. [TLS07] propose a bilingual LSA framework which enforces a one-to-one topic correspondence between source and target models. This allows for the target topics inferred on the source sentence to be used for target language model adaptation in one decoder pass. They also perform translation lexicon adaptation by training topic-dependent translation lexicons and adding two LSA-adapted phrase scores to the phrase table.

Language models only have a narrow context, usually the three or four preceding words, to predict the next word in a sentence and are not ideally suited to the task of disambiguating homonyms. Even using topic-dependent n-gram probabilities, they still have very little information on which to decide which meaning of a homonym is the correct one in a certain case. By adapting the translation model we can directly influence the selection of phrases and with it the translation candidates considered by our decoder. The following approaches all adapt the translation model directly using topic information.

Zhao and Xing [ZX08] use a Hidden Markov Bilingual Topic AdMixture (HM-BiTAM) to learn word alignments dependent on locality of the source words and their topical context. In their bilingual model a source word depends on the topic, whereas a target word depends on the topic and the aligned source word. Using this model they can estimate topic-dependent translation lexicons and unigram frequencies. Similar to [TLS07], a mixture of topic-specific translation lexicons is used to score phrase pairs in the decoding process.

Gong et al. [GZZ10] take a different approach and dynamically filter the phrase table according to topic information. They trained a monolingual LDA topic model on the source-side of their corpus. During the extraction of phrase pairs, phrases are annotated with document information. The phrase pair is assigned the most likely topic from all the interpolated document topic distributions associated with it. If the probability of the most likely topic is under a certain threshold, the phrase pair is considered to belong to no particular topic. During decoding, only phrase pairs which match the topic of the source sentence and "general" phrase pairs, belonging to no particular topic, are used. If the source sentence itself does not belong to one particular topic, all phrase pairs are used. While their experiments show an average improvement in BLEU score of over one point on a Chinese-to-English translation task, their results are reported on a few hand-selected documents spanning only three of the ten topics they trained, leaving open the performance of the system on the other seven topics.

In one of the most recent works in the field, Xiao et al. [XXZ$^+$12] developed a topic similarity measure for hierarchical PBMT. In hierarchical PBMT, phrases of phrases can be learned by extracting phrases that contain placeholders. This allows modeling of discontinuous language constructs, such as for example the French negation "ne ... pas", and facilitates long-range reordering. Xiao et al. assigned each synchronous rule a topic distribution. This is done by interpolating the source and target-side topic distributions of the bilingual corpus from which the rules are extracted. Since only source-side topic distributions are available during decoding, the learned target-side rule-topic distributions are projected into the source-side topic space using a one-to-many topic correspondence. To decide which rules should be applied during decoding, they calculated the similarity between the source document topic distribution and the rule topic distribution. If a topic distribution has a high entropy, assigning equal probability to many topics, it is categorized as a topic-insensitive rule and applied to all documents.

3.1 Our Work

In this thesis we propose a new approach to incorporating topic models into machine translation. While language model adaptation has been widely studied in the fields of ASR and SMT, research on topic adaptation in machine translation is still a rather young field and few works have investigated ways of incorporating topic models directly into the translation process. One of the main advantages of a log-linear model in machine translation is its inherent modular structure enabling easy inclusion, exclusion and modification of various components. We can lever this advantage by adapting a component that models the translation dependencies between words directly. The discriminative word lexicon is a component in the log-linear model that guides the choice of target words given the source sentence. In this thesis we present a new method of topic adaptation in machine translation by combining monolingual topic models with a DWL. Unlike a language model, the DWL has a global view

We tried to find a way to implement topic adaptation in the translation process that can be easily incorporated into existing frameworks. While hierarchical PBMT used by Xiao et al. [XXZ$^+$12] has been shown to work well on translations from Chinese to western languages, it is not used in many decoders built for translation between western languages. Gong et al. [GZZ10] filter the phrase table to only apply phrases with a matching topic. This approach involves directly changing the decoder code and only works on large corpora with few topics to avoid data sparsity problems. Translation lexicon adaptation as used by [TLS07] and [ZX08] seems promising, but requires the training of complex bilingual topic models and a change in the decoder code to include new lexical weights for phrases.

The DWL is a form of translation lexicon that does not work on the phrase level, but rather utilizes information at the sentence level. It uses only source sentence words as features. We can therefore train a monolingual topic model only on the source texts and do not have to worry about target language topics. This simplifies training of the topic model, as monolingual topic models have been well-studied and many implementation toolkits are available on the web. The topic model may also be reused when translating the same source text into a different target language, cutting down on training costs in scenarios where a source text is translated into many languages. One of the main advantages of our approach is that it is easy to implement. The phrase extraction and decoding process remain largely unchanged, making it universally applicable. A topic-enhanced DWL can simply be added as an additional feature in the log-linear model. This allows easy integration of topic adaptation into existing machine translation systems.

An important point to consider in topic modeling is the necessity for document boundaries. Most work done in topic adaptation is based on the assumption that the document boundaries of the corpus are known. However, many of the available large parallel corpora do not contain any document boundary information. Ruiz and Federico, who worked on the TED corpus, which includes talk boundaries, chose to segment the corpus into documents of 5 lines length instead to "simulate near-time translation of speeches" [RF11]. Gong et al. segmented the FBIS corpus into documents of varying length using the TextTiling algorithm, which automatically clusters paragraphs into documents on the basis of subtopics, followed by a manual inspection to correct segmentation errors. Although the TED corpus we use in this work contains talk boundaries, we investigate multiple ways to divide the corpus into documents that might be applied to corpora without document boundary information. We note the impact of document size on topic modeling and translation quality.

4. Approach

This chapter focuses on our approach to incorporate topic models in the translation process. We explain how we incorporated topics as source features into the DWL as a three step process. We give an account of how first the topic model and second the DWL is trained, before describing the typical process we underwent for every new topic configuration. Following this, we conclude the chapter with a discussion of the parameters we studied in our experiments.

4.1 Incorporation of Topic Models

In order to incorporate topic models into the decoder, we utilized the topics as source features in our DWL. This enables the DWL to consider the context in which the source sentence appeared when deciding whether a target word should be used in the translation of the source sentence. No time-consuming and error-prone changes in the source code of the decoder have to be performed beyond the inclusion of a DWL in the log-linear model.

To augment our DWL with topic information, we first prepared our corpus. The topic model operates on a document level. Although the TED corpus contains talk boundaries, we also investigated the performance of our model on smaller document sizes. Segmenting the corpus into shorter documents is also necessary whenever no document boundary information is present in the corpus. The documents were then preprocessed similar to the standard preprocessing in our baseline system.

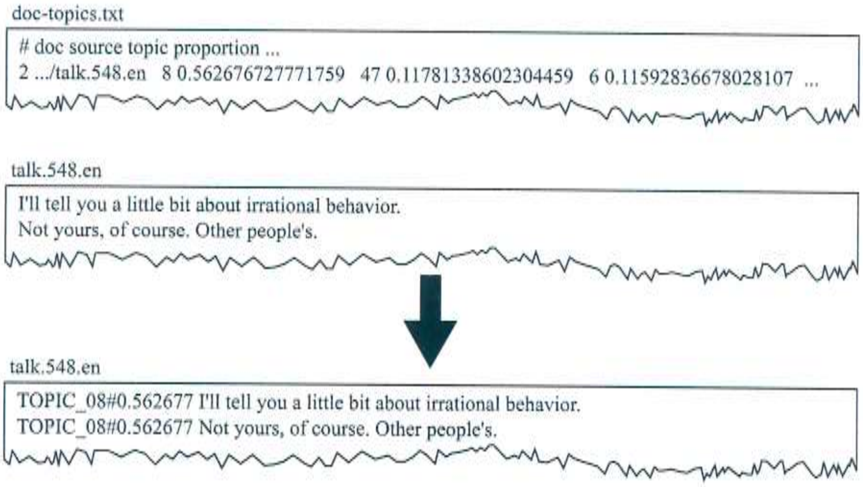
We trained a monolingual topic model using Latent Dirichlet Allocation on the source side of our trainings corpus. The topic model was then used to infer topics on the documents of the development and test set. For every line in a document, we inserted a topic marker representing the most likely topic from the document topic distribution output by the model. This topic-augmented data was used to train a DWL. Finally we ran a translation on the topic-augmented source sentences to evaluate our DWL.

4.1.1 Training a Topic Model

In this work we used the *Machine Learning for Language Toolkit* (MALLET) [McC02] for all topic models. MALLET was developed by Andrew McCallum at the University of Massachusetts Amherst. It is an open source software written in Java that includes support not only for topic modeling, but also document classification, sequence tagging and other natural language processing applications. MALLET implements topic modeling with Latent Dirichlet Allocation using Gibbs sampling. We used MALLET because it is platform-independent, freely available and supports output of detailed topic statistics useful in analyzing and manually inspecting topic models.

Before we imported the data into MALLET, basic preprocessing was performed. This included part of our usual preprocessing, such as splitting off punctuation and padding special symbols. We also split up the corpus into smaller documents for topic modeling, as explained in detail in Section 4.2.5. Stop word filtering and tokenization was done during the import of data into MALLET. We evaluated a number of different stop lists as laid out in Section 4.2.4.

After the data was imported into MALLET, we trained a topic model using configuration-specific parameters. The trained models were written out and, if necessary, manually inspected. We then used the trained topic model to infer topics on the development and test set. The inferred topics were incorporated into the data through the use of topic markers. For each document the topic model outputs a document topic distribution. We added up to five topic markers in each line, representing the most probable topics for this document from the document topic distribution. In some configurations, we also included the probability for the given topic in the topic marker.



doc-topics.txt

```
# doc source topic proportion ...
2 .../talk.548.en   8 0.562676727771759   47 0.11781338602304459   6 0.11592836678028107 ...
```

talk.548.en

```
I'll tell you a little bit about irrational behavior.
Not yours, of course. Other people's.
```

talk.548.en

```
TOPIC_08#0.562677 I'll tell you a little bit about irrational behavior.
TOPIC_08#0.562677 Not yours, of course. Other people's.
```

Figure 4.1: Exemplary augmentation of a TED talk with topic information. The first document shows an excerpt from the trained document topic distribution, listing the source document and the probability for each topic ordered by probability. The middle document represents the first couple of lines of the corresponding TED talk. The lower document shows the same lines of the talk after it was annotated with topic markers denoting the most likely topic and its probability. Every line in the document was annotated with the same marker.

Figure 4.1 shows how we annotated a TED talk with topic markers. The first document is an excerpt of an exemplary document topic distribution output by MALLET. Each line contains the topic distribution for one document, ordered by the most likely topic being first. In the example each talk was used as one document for topic modeling, and the topic model was trained using 100 topics. Also shown in the figure are the first lines of the talk before and after it was annotated with topic information, in this case the most likely topic and its probability. Every line in the document was annotated with the same topic marker. While the figure shows an example using English text for ease of understanding, all topic models were trained on the German source texts and only the German side of the parallel corpus was annotated with topic markers.

To keep training conditions consistent between the baseline system and our new configurations, we added the topic markers for the inferred topics to the regularly preprocessed documents. The topic-annotated documents were then concatenated to form the actual training, development and test corpora.

4.1.2 Training a DWL

The topic-annotated German training data was used in conjunction with the original English data as the parallel corpus on which we trained a DWL. For this process the topic markers were treated as any other word in the source vocabulary. In a first step, the phrase table was pruned to contain a maximum of ten phrase pairs per source phrase the same way it was done during decoding. Given a single source sentence, all phrases from the phrase table that could be applied to this sentence were gathered. The target words of these phrases constitute all the words that might possibly occur in a translation of the source sentence produced by our decoder. The DWL operates only on the set of source and target words. As a consequence, both the matching target words and the source sentence words were stored in a bag-of-words format. This is to say every word was counted once, irrespective of the number of times it occurred in one line, and the order of words was disregarded. This was done for all sentences in the training set.

For the development and test data, a target word vocabulary was constructed using the target words from the phrase table. We then trained a binary maximum-likelihood classifier for every target word in the vocabulary using the *MegaM* toolkit [Dau04]. It learns whether a target word should be included in the translation given the source sentence. We used the MegaM software because it provides a fast and efficient implementation of maximum entropy models.

Following [MCN+11], we trained the DWL merely on a subset of the corpus. Given a single target word, only the source sentences for which the target word was part of the matching phrases were considered. In other words, we only looked at sentences that contained the target word in their list of translation candidates. This avoids training the target word on source features which will never occur in the calculation of a score and results in sharper models. As an added benefit, less training time is needed. If the given target word occurs in the reference translation, the words of the source sentence are used as positive features in MegaM. Otherwise, the source words are counted towards the negative class.

After training of the weight files, the DWL score of a single target word can be computed as follows:

$$p(e|\mathbf{f}) = \frac{1}{1 + e^{-(\text{bias} + \sum\limits_{f \in \mathbf{f}} \lambda_{ef} \cdot w_f)}} \tag{4.1}$$

where λ_{ef} is the feature weight learned by MegaM and w_f is the corresponding feature value. The score for the whole target sentence is then easily computed as the product over all target words as laid out in equation 4.2.

$$p(\mathbf{e}|\mathbf{f}) = \prod_{e \in \mathbf{e}} p(e|\mathbf{f}) \tag{4.2}$$

The topic markers were treated as any other word in the computation of the DWL score. The only exception was when we used the probability of the topics as an additional information source. In these cases, we used the probabilities as feature values. A more detailed description of the process is given in Section 4.2.3.

4.1.3 Running a New Configuration

For every new configuration, we ran through the steps of training a topic model and DWL as laid out above. To evaluate a new configuration, we ran a translation using the topic-annotated source test data and the original target test data. The newly trained topic DWL replaced the DWL of our baseline system. Apart from the DWL all the components from the baseline system remained unchanged. For the most part, we did not perform any MER training on the topic system. Instead, we reused the log-linear model feature weights that had been optimized on the baseline system.

For the main part of our experiments, no changes had to be made in the source code of the decoder. The topic markers were only used in the computation of the DWL score for every hypothesis and did not influence the other models in our system.

4.2 Topic Model Parameters

Before we can train a new topic DWL, there are a number of parameters to consider, such as the number of topics in the topic model and the size of the documents on which the topics are trained and inferred. Since topic modeling is a statistical process, no two trained models are alike even if exactly the same parameter configurations are used. Despite this uncertainty, significant variations in the quality of the topic models and the final translations can be observed when the parameters are changed. The success of our approach hinges on a well-performing topic model, which in turn depends on these parameters. In order to evaluate the true potential of our approach, we had to find the optimal value for every parameter.

In this section we introduce the parameters we examined in our experiments. Table 4.1 provides an overview over the parameters and their corresponding values. The number of parameters and their possible values do not allow an exhaustive evaluation of all possible combinations. Given the time constraints of this thesis, we were only able to investigate a few chosen configurations. To narrow down the number of options, we first tried to determine the optimum for a couple of parameters, which we then held constant for all subsequent experiments. This section is only meant to introduce the parameters we studied. An in-depth discussion of their effects and translation results for each parameter are given in Chapter 5.

parameter	values
number of topics	10, 100, 1000
number of topic markers	1, 3, 5
include probability	yes, no
segmentation of texts	talks, 40 lines, 10 lines, sliding window, single lines
stop lists	MALLET default, TED-specific
MERT weights	optimized, baseweights

Table 4.1: Overview of the parameters for topic modeling and their possible values which we evaluated in the different experiment configurations.

Apart from the parameters listed in Table 4.1, we also investigated a few special configurations which are described in detail in Section 5.8.

4.2.1 Number of Topics

As discussed in Section 2.3.1, LDA requires the number of topics to be known and fixed before a topic model is trained. This is arguably one of the most important parameters when training a topic model. Choosing too few topics would result in too broad, unspecific topics, whereas too many topics do not generalize well and may cause data sparsity problems. It was therefore crucial to choose a reasonably well-fitting number of topics early on. To gain insight into how different topic models behave given our corpus, we trained and manually evaluated multiple topic models with a varying number of topics. From these, we chose three models and tested their performance on our corpus. Translation results of topic models with 10, 100 and 1000 topics are presented in this thesis. Based on these results, we chose the best number of topics for all subsequent configurations.

4.2.2 Number of Topic Markers

The idea behind topic modeling is that a document consist of a mixture of multiple latent topics. This idea is reflected in our document topic distributions. Most documents will exhibit a high probability for a small number of topics and a low probability for all other topics. If we only include a topic marker for the most probable topic in our source sentences, we disregard the other topics inherent in the document. Including too many topic markers on the other hand bears the risk of including topics that are not very relevant for our document and might only confuse matters. Another factor to keep in mind is the overall number of topics in our model. If we only have 10 topics, including the 3 most probable topics in every talk is probably less helpful than if we had 1000 topics in our model. To evaluate how many topic markers should be used, we ran translations using 1, 3 and 5 markers for the most likely topics of each document.

4.2.3 Including Probability

So far, we have only looked at the document topic distribution to find out the most likely topics. The topic distribution for a document contains another kind of information we might be able to utilize in our model: the probability for each topic. A high probability indicates that the topic is rather dominant in the document and many words in the document were drawn from this topic. It might also be seen as a kind of confidence score denoting how prevalent the topic is in the document and therefore how strongly we should rely on it. Probability values might be especially useful when we include more than one topic marker in our source documents to distinguish between the likelihood of the topics.

We therefore tried to lever this information by including the actual probability of the most likely topics into our topic-augmented documents. The features used to the train the DWL are not weighted. The DWL was therefore trained as usual on the normal vocabulary including topic markers but no probabilities. To factor in the probability in the DWL, we modified Equation 2.6 given in Section 2.2. Instead of using binary features, we used the following feature weight equation:

$$\phi(f, \mathbf{f}) = \begin{cases} 1 + prob(f) & \text{if } f \in \mathbf{f}, \\ 0 & \text{else} \end{cases} \tag{4.3}$$

In case of a topic marker, $prob(f)$ is the probability of the topic in the topic distribution. For all other words, it is 0. This weighed our topics more strongly than the normal German words and emphasized more likely topics over less likely ones. In order to implement this change, we had to perform some minor modifications regarding the calculation of the DWL score in the decoder source code.

4.2.4 Stop Lists

When modeling topics, we are not interested in function words such as articles, pronouns, auxiliary verbs, prepositions or conjunctions. They carry little to no content information and their translation does not rely on topical information, but rather on their syntactic context. We can assume for them to be equally distributed across all topics and might thus be inclined to ignore them. This is a valid approach whenever the document size on which topics are modeled is rather small and in many of our configurations we do not bother with stop word filtering beyond the default, very small stop list provided with the MALLET toolkit.

However, as function words by their nature appear with a higher frequency in a text than content words, stop word filtering becomes more important when dealing with longer documents. Given sufficiently large documents, function words can skew the topic model. In these cases the *stop words* have to be filtered out prior to topic modeling. This requires great care. Filtering out too few words can result in a generic topic which is very likely for every document topic distribution, because function words appear in every talk regardless of the subject of the talk. Filtering out too many words, on the other hand, might remove valuable topic clues from our data.

We built multiple stop lists by gathering the most frequent words in two different corpora. We used the TED talks as a task specific corpus and a large corpus of German newspaper articles as used in the 2012 Workshop on Statistical Machine Translation [CBKM$^+$12] as a general corpus. Some stop lists contained only words from one corpus, whereas others combined the most frequent words of the two corpora. The stop lists generally differed in length. We used the stop lists to filter our source texts prior to topic modeling. The resulting topic models were manually inspected for their performance on the unseen test set. Of the evaluated stop lists, only one worked well on whole talks as documents for topic inference. Translation results using this stop list for topic inference on whole talks are reported in the following chapter.

4.2.5 Segmentation

Document size is an important factor in topic modeling. LDA operates on the document level by inferring the most likely latent topic distribution that might have created our document. This means we cannot simply apply topic modeling to our whole corpus, or we would end up with only one topic distribution. Instead, we always need to find a way to segment our corpus into smaller documents so that we end up with one document topic distribution for every document in our corpus.

The talk boundaries are known for the TED corpus, which means we know when a talk ends and a different one begins. Using this boundary information, we might simply use each talk as a document for topic modeling. However, this intuitive approach might not be the best choice. TED talks are comparatively short, with the longest talks lasting about 20 minutes or a couple hundred sentences. But even given such a short time, a talk might touch upon different topics. What if we were translating university lectures of 90 minutes instead of short talks? We might still have the boundary information of a lecture, but we would certainly choose a smaller size for our documents on which to perform topic modeling. This shows that even though our corpus contains document boundaries, it can be worthwhile to investigate different segmentations. Furthermore, for many widely-used corpora document boundary information is not readily available at all. To still be able to apply our approach in these cases, the corpus needs to be split into smaller documents. This can be done either arbitrarily or knowledge-based. Given the importance of documents for topic modeling, we chose to investigate the effects of different document sizes on the performance of our system.

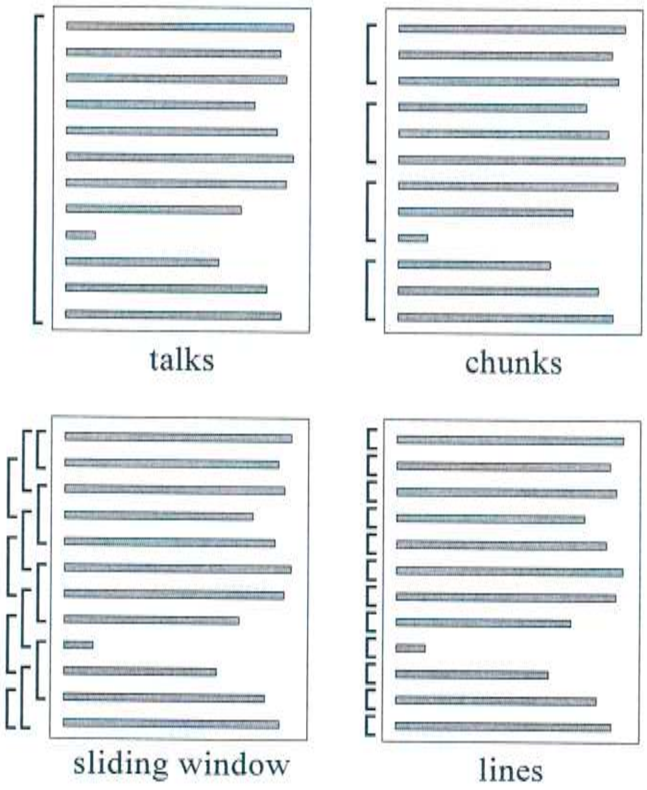


Figure 4.2: Illustration of the different methods we employed to segment a talk into documents for topic modeling. The left hand side brackets indicate which lines were grouped together into one document.

We explored four different ways of segmenting our corpus into documents for topic inference: talks, chunks, sliding windows and single lines. Figure 4.2 illustrates the different methods we employed using the example of a single TED talk. For every segmentation method, the left hand side brackets indicate the lines which were grouped into a document. One side effect of varying document size becomes immediately apparent. The size of our documents is inversely proportionate to the number of documents and therefore also to the number of document topic distributions. This might have an impact on training time, especially for large corpora.

We did not always apply one segmentation method to the whole corpus, but also investigated the effects of discrepancy between training and inference document sizes. During training we have all the data for our system readily available and do not have to limit ourselves. Depending on the translation task, we might not have that luxury in testing. In a simultaneous or near-simultaneous translation task, we might only have the last few utterances at our disposal. We investigated whether its possible to use a topic models trained with a larger context on our training corpus to successfully infer topics on a small amount of testing data.

Segmenting our corpus into talks or single lines is rather straightforward. We simply considered each talk or every line as a document. While the segmentation into talks is intuitive, it is also quite coarse and does not allow for topical variation over time. On the other hand, it would not be sensible to learn a topic model using single lines as documents, since we use topic models in order to lever information beyond the scope of a single sentence. However, we might still successfully apply topic models learned on larger documents to single lines in testing. The segmentation into single lines was therefore only used on the development and test datasets.

The other two methods we employed to partition our datasets, chunks and sliding windows, are described in the following sections.

Chunks

The longer a document, the greater the chance that its topical content will change over time. A 20 minutes long talk will often contain multiple sections, each of which might cover a different topic. To account for the changing topics over time, we decided to segment our corpus into documents with a fixed number of lines. In our experiments, we used chunks with a length of 10 or 40 lines each. The length of TED talks varies considerably, spanning from only ten at the lowest to several hundred lines at the most with an average length of 133 lines across all datasets. We chose 40 lines as a very rough approximation of segmenting an average talk into three parts - beginning, middle and end. On the other hand we felt that 10 line documents were just long enough to give us sufficient context for training a topic model while still allowing us to infer topics on single lines, simulating an on-line translation task.

Document boundary information present in a corpus might still be used in conjunction with segmenting a talk. We used two different ways to chunk our datasets, once respecting boundary talk information and once discarding it.

Given the talk boundary information, we can simply segment each talk, so every chunk only contains lines from the same talk. This will result in some documents that are only a few lines long containing the last few lines of a talk. In the worst case, we would get a talk only a single line long containing all of two words: *thank you.* To avoid these short segments we balanced the length of our documents by concatenating every chunk shorter than half of the desired length with the previous chunk. Given 10 line chunks, this resulted in chunks between 5 and 14 lines in length.

Assuming we do not know any document boundaries for our corpus, our datasets would consists of all the appropriate talks concatenated together. We then took these datasets and spliced them after every 10 lines. As an advantage, all except possibly the last chunk will have the same length. However, this approach will produce chunks that span talk boundaries, consisting of both the last lines of the previous talk and the first few lines of the next talk. How gravely this affects topic inference depends on the size of chunks and type of source material. The first and last lines in a presentation often contain little topical information, but are filled with introductions and thank yous. Manual inspection of topics models learned from this approach showed that they usually contained one topic that would assign a high probability to *thank*, *you* and *applause*. Boundary-spanning chunks would often be assigned this "end of talk" topic.

Sliding Windows

In the previous methods, we would partition our datasets into documents, and all the lines in the document would be annotated with the same topic markers for the most likely topic of the document topic distribution. In these approaches it is completely arbitrary whether a particular sentence is counted towards the previous or the following sentences for our topic model. This creates harsh borders for topic changes. To soften these, we adapted a sliding window approach. While talk boundaries may be respected in the same way as our chunking approach, the sliding windows can also be applied to a corpus without any boundary information.

We chose a window size of 10 lines because early experiment results on 10 line chunks looked promising. For every line in the corpus or in a talk, depending on whether talk boundary information was utilized, we created a document containing the previous 5 and following 4 lines. For the first line of the corpus, the document would contain this line and the following 4. Similarly the document for the last line would only contain the last 6 lines of the talk. This means all our documents were between 5 and 10 lines long.

We used these documents to train and infer the document topic distributions. Instead of adding the most likely topic to every line in the document, we added the topic marker only to the line for which the document was created. We then built our training corpus for the decoder by concatenating every line with a topic marker. In the resulting topic-augmented corpus the topic might change at every line, in the same way if we had inferred the topics on single lines. However, unlike using single lines, we utilized the surrounding context of that line to infer the topics.

4.2.6 MERT Weights

The DWL is one of many components in our log-linear model, each of which has its own feature weight. The weights determine how much influence a component has on the final score of a hypothesis. In many state-of-the-art systems, the feature weights are optimized iteratively with regard to an automatic error metric (most commonly the BLEU score) using *Minimum Error Rate Training* (MERT) [Och03].

We have chosen to only optimize handful of configurations for several reasons. When we use a topic DWL instead of our regular baseline DWL in our decoder, we would normally optimize the system with the topic DWL to achieve the best score. However, this might skew our results as we expect the impact of topic models on the BLEU score to be relatively small. We would not be able to tell anymore whether the difference in BLEU score between the topic system and our baseline was due to our improved DWL or resulted from the changed log-linear feature weights. We therefore run all of our configurations with the weights obtained from optimizing the baseline system (baseweights). This way even a small difference in BLEU score can be traced back to our improved DWL. Lastly, it should also be kept in mind that MER training is quite costly as it requires a significant amount of time and computational power. Foregoing MER training allowed us to quickly analyze new configurations.

However, running all configurations only on the baseweights would not give accurate results of the true performance using topic models. While they are quite similar, the topic-augmented system uses different features than our baseline. It might therefore exhibit a higher predictive power in choosing the right words in the translation process than a normal DWL. We would expect MER training to afford the topic DWL a higher feature weight, possibly at the expanse of the language model. To see the full potential of our systems, we used MER training with topic-annotated development data on a few choice configurations. We also used MER training for all special-case configurations, because they used DWLs that differed significantly from the baseline DWL.

5. Evaluation

In this chapter we provide an overview of the TED corpus, the data we used for the main part of our experiments, and then briefly describe the design of our baseline system. Afterwards we present our findings from evaluation runs using different topic configurations on the TED corpus. We examine the impact of the topic parameters introduced in the previous chapter on the translation result and discuss the challenges that arose in the performance of the experiments. Afterwards, a number of unique configurations outside the usual parameter range and their results are reviewed. To further examine our results, evaluation measures beyond automatic evaluation metrics are analyzed. We discuss selected translation examples and their impact on the system quality.The chapter concludes with the presentation of an additional translation task, where we evaluated our approach on the German to French translation task from the Quaero 2012 evaluation.

The experiment results in this chapter are reported using two of the most commonly used automatic evaluation metrics: the *Bilingual Evaluation Understudy* (BLEU) [PRWZ02] and NIST [Dod02]. We used BLEU$_4$ scores, which have been computed using n-grams of a length up to four. Both case-sensitive and case-insensitive scores are reported. All scores are computed using a single English reference sentence and contain punctuation marks.

5.1 Data

In this work we used the German-English TED talks[1] set from the *Web Inventory of Transcribed and Translated Talks* [CGF12]. TED talks have been used in the past most notably in the machine translation track of the evaluation campaigns of the *International Workshop on Spoken Language* (IWSLT) since 2010.

The TED corpus is a video collection of public educational talks that is freely available from the TED website under the Creative Commons Attribution – NonCommercial – NonDerivative 3.0[2] license. The talks have been given at conferences around the world organized by the nonprofit organization TED. TED stands for Technology, Education, Design and subsequently the talks range over a whole number of subjects from technology, science or global issues to entertainment. Most talks are fairly short, usually between three and twenty minutes. They are mostly given in English, although many speakers are non-native. The talks have been professionally transcribed and translated by volunteers in more than 90 languages.

[1] `http://www.ted.com/talks`

[2] `http://creativecommons.org/licenses/by-nc-nd/3.0/`

set	#talks	#lines	words	
			German	English
train	384	51,021	761,544	821,726
dev	7	796	13,506	14,602
test	11	1,649	25,684	27,805

Table 5.1: Overview of the TED corpus.

Table 5.1 shows the statistics of the corpus as we used it. The corpus is case-sensitive and includes punctuation. It has been partitioned into three disjoint sets of talks. The training set is used to train all our models, which are then tuned on the development set. Lastly the test set contains unseen talks used for the evaluation of our models. As far as machine translation corpora go, the TED corpus is rather small. It only contains roughly 800K words as opposed to large corpora such as the Europarl corpus [Koe05], which contains well over 25M words per language. The TED corpus is still growing, as new talks are published and translated on the TED website weekly.

The original TED data consists of transcripts that are broken up into utterances for the subtitle frames, most of which do not correspond with sentence boundaries. Many frames contain only fragments of a sentence, but may also at other times span sentence boundaries. Our machine translation system translates one line at a time. Since our goal is to translate full sentences, the parallel sets of utterances have been automatically concatenated to approximate sentences. This was done by joining consecutive utterances until they ended with a punctuation mark. We did not split utterances if they contained punctuation marks in the middle, because there is no strict sentence correspondence. One English sentence might sometimes be translated as two German sentences or the other way around. To determine whether this was the case would have required a proper word alignment. Rather than risk chopping off half a sentence, we erred on the side of caution. As a result every line of our joined corpus contains at least one sentence. Minor error correction was done by hand on the development and test set.

Every talk in the TED corpus is annotated with meta-data, such as the speaker, title and a description of the talk. This data has not been used with the exception of keywords, which were used in one special configuration run (cf. Section 5.8.1). The keywords are used as tags on the TED website[3] and have been assigned to the talks by hand. Although most of the talks are tagged, some talks are missing these keywords.

The TED corpus presents a unique challenge for machine translation. Unlike news articles or books which have been carefully worded, the TED talks contain a mix of well-planned and spontaneous speech depending on the speaker and situation. Speech is inherently more difficult to translate than written texts, because it contains disfluencies such as false starts and repetitions and generally exhibits a greater variance. A main feature of the TED talks is the variety of subjects they cover, which also makes the translation more difficult than a narrowly defined translation domain. On the other hand it is our hope that topic models will prove especially useful in an relatively open-domain setting. The TED translation task is also closely related to that of lecture translation.

[3] `http://www.ted.com/talks/tags`

5.2 Baseline System

As our baseline system we used a very basic state-of-the-art phrase-based decoder with short-range reordering lattices and a Discriminative Word Lexicon.

Before training the models, we used standard preprocessing techniques. Excessively long lines were removed from the trainings set, as well as lines that were the same in both languages. This reduced the training set from 51021 to 47338 lines. During preprocessing, we also normalized numbers and dates. The first word of every sentence was smart-cased. Punctuation marks were split off with whitespace and common abbreviations and contractions were written out. The German training set was normalized with regard to the German orthography reform of 1996 (neue deutsche Rechtschreibung) and compound-splitting was performed.

After preprocessing, we trained the word alignments in both directions with the GIZA++-Toolkit [ON03]. They were symmetrized using the grow-diag-final-and heuristic. From this alignment the phrases were extracted using the Moses toolkit [KHB+07]. We trained a DWL on the training set and filtered it for phrases occurring in the development and test set as explained in Section 2.2. A 4-gram language model was trained with the SRILM Toolkit [Sto02] on the target side of the TED corpus using Kneser-Ney smoothing. The system was tuned on the development set using Minimum Error Rate Training (MERT) [Och03] with regard to the BLEU metric [PRWZ02].

For each sentence on the source side, short-range reordering was performed with a POS-based distortion model [RV07]. To accomplish this, the source side was first tagged with the TreeTagger [Sch94] to generate part-of-speech labels. Reordering rules were learned on the word-aligned parallel training data. Given the monotone lattice of a source sentence, matching reordering rules were then applied by introducing new edges into the lattice. The result is a translation lattice containing multiple possible reorderings of the source sentence. The paths of the lattice can then be monotonously decoded. Our hypotheses were constructed using these lattices, but the DWL scores were computed on the original source sentence. This allowed us to augment our source text with topic markers in later experiments without including them in the final translation hypothesis output by our decoder.

During decoding all possible phrases that match the source sentence were inserted into the reordering lattice, resulting in the final translation lattice. The hypotheses were built using the KIT beam-search decoder [Vog03] and finally the output was evaluated using automatic evaluation metrics.

5.3 Number of Topics

LDA requires the number of topics to be set before the topic model is trained. Finding a good number of topics to be used is critical for topic model performance. Given too few topics our model will not possess enough discriminative power, given too many topics our model will be overfitted and unable to generalize. Our first experiments therefore centered around finding a good number of topics to use in all subsequent configurations.

Before running any configurations, we trained and manually inspected a number of topic models. During these early tests it became apparent that using small chunks of 10 lines each as described in Section 4.2.5 resulted in sharper topics than using whole talks as documents. We therefore trained our initial topic models on documents of 10 lines each. To gain insight into how the topic models performed in conjunction with a DWL, we ran translations using three different configurations.

number of topics	case-sensitive		case-insensitive	
	NIST	BLEU	NIST	BLEU
baseline	6.310	23.05	6.445	24.12
10	6.314	23.08	6.449	24.16
100	**6.324**	**23.13**	**6.457**	**24.20**
1000	6.296	22.95	6.429	24.01

Table 5.2: Experiment results regarding the performance of topic models trained on a varying number of topics.

Table 5.2 shows the results we obtained by running configurations with 10, 100 and 1000 topics. The topic models for all three configurations were trained on 10 line chunks of the training data and topics were inferred on single lines of the test data. The default MALLET stop list was used for all topic models. For DWL training, the most likely topic was added as a topic marker to the source text. The baseline includes a DWL trained without topics. As feature weights for the log-linear we re-used the weights from the baseline system, which has been optimized using MER training.

The configuration using 1000 topics performed the worst of all three, achieving a score lower than our baseline system. This is expected, as it's a huge number of topics – especially for a small corpus. There is not enough training data to adequately train the topics and thus they perform poorly. We also cannot expect 1000 topics to be prevalent in our corpus. Thus we did not take 1000 topics into further consideration.

10 and 100 topics are topic numbers much more commonly used in related works on topic modeling and, as can be seen in Table 5.2, they worked much better. Both configurations

10 topics			100 topics		
welt	haben	zeit	welt	buch	universum
menschen	um	noch	sehen	also	raum
haben	diese	jahren	länder	wirklich	draußen
kinder	dann	hat	indien	schreiben	materie
prozent	noch	ihnen	china	sprechen	also
point	mehr	wurde	staaten	lesen	masse
mehr	weil	zwei	vereinigten	sprache	auge
dollar	viel	wurden	land	bücher	physik
millionen	diesen	um	ländern	wort	bewegen
heute	damit	jahre	unten	sehr	stellen
geld	dabei	ohne	usa	wissen	anderen
jahr	diesem	paar	gesundheit	geschrieben	universums
land	dazu	alle	zwischen	wörter	befinden
pro	genug	während	europa	möchte	dann
probleme	gerade	ersten	schweden	schrieb	entdecken

Figure 5.1: Three examples each of German topics generated from the TED corpus using topic models with 10 and 100 topics.

using fewer topics achieved a score better than the baseline, improving our system. The configuration trained on 100 topics achieved the best BLEU score, improving our baseline result by 0.08 BLEU points. While this is a marginal improvement, we have left all the factors contributing to our score the same except for the DWL. We can therefore be sure that this slim improvement in BLEU score stems directly from the introduction of topic models into our DWL.

Comparing the configurations using 10 and 100 topics, they both performed reasonably well. While 100 topics achieved the best results, this might not hold true for different document sizes. In order to choose the better model, we manually looked at the 10 top words for each topic in the models. Figure 5.1 shows three examples of topics trained on the German source side of the corpus for both topic models. The topics from the model using only 10 topics were rather unspecific and not easily classified as pertaining to a certain subject. On the other hand, the topics from the model trained on a 100 topics showed a nice coherence between the words for one topic and could for the most part easily be labeled. Although both models were trained on small documents of 10 lines each, the 10 topics showed a much higher concentration of function words than the 100 topics. We felt that 100 topics were better suited to discriminate between different homonyms. Given this and the fact that it was the best performing model in our first experiment, we chose 100 topics as the topic number for all subsequent configurations.

5.4 Multiple Topic Markers and Probabilities

So far, we have only used the most probable topic as a topic marker in our source texts. However, the intuition behind topic models is that a document exhibits a mixture of different latent topics. If we only use the most probable topic, we are discarding possibly valuable information of our topic model. We tried to leverage this information by including multiple topic markers for the top most likely topics in our source sentences.

When we use multiple topic markers, for example the three most likely topics, we do not know how likely they are. For one talk the top three topics will all be equally probable, while for another talk only the first topic will have a high probability and the next two topics will be considerably less likely than the first one. We therefore looked not only at the number of topic markers, but also at the probability of a topic as an additional information source for our DWL. We used the probability as a feature value in the calculation of the DWL score as explained in Section 4.2.3. This required a small modification in the source code of the decoder. Apart from that, the DWL was trained as usual.

We used the same topic model for all four configurations, which is the same topic model trained on 10 lines chunks and inferred on single lines that was used in the previous section.

# of topic markers	probabilities	case-sensitive		case-insensitive	
		NIST	BLEU	NIST	BLEU
baseline		6.310	23.05	6.445	24.12
1	no	**6.324**	**23.13**	**6.457**	**24.20**
3	no	6.311	23.06	6.445	24.12
1	yes	6.310	23.07	6.445	24.15
5	yes	6.312	23.03	6.448	24.09

Table 5.3: Experiment results using multiple topic markers for each document and the associated probability for each topic as feature values for the DWL.

This model was chosen because it has shown the best translation results in our previous experiment. We trained a new DWL for every configuration using more than one topic marker or probabilities. Table 5.3 gives an overview of the results we achieved in our experiments. The configuration using only one topic marker for the most likely topic and no probability information performed the best out of all configurations we tested.

Contrary to our expectation, using the three most likely topics instead of only the most likely one hurt the performance of our system. Instead of helping with disambiguation, the additional topics confuse matters further for the model, diminishing its discriminative power. The same effect can be observed when we use probabilities. The system using only the most likely topic performs better than the one using multiple topic markers even with the additional weighting of the topics.

The problem with using multiple topics is that their influences on the DWL score can cancel each other out. While one topic will favor the correct translation over the wrong baseline phrase, another topic will favor the incorrect baseline translation. When we then use both topic markers in our source sentence, the negative and positive scores of both topics add up close to zero. As a result we still produce the baseline hypothesis, although we would have had the information to correctly disambiguate the word available. The same effect can be seen when we use keywords instead of topics as explained in Section 5.8.1.

We used the five most likely topics instead of only three for the configuration using probabilities expecting that the model would weigh the topics according to their probability. This means less likely topics should not be considered as much as the most likely topic by the DWL. However, the configuration performed worst out of the four configurations we tested, showing that even the weighting of multiple topic markers does not help in making the correct translation decisions.

While the configuration weighing the single most likely topic marker with its probability performs better than our baseline system, it does not manage to outperform the configuration using no probabilities at all. We therefore decided against using probabilities as feature values in our further experiments. Since it has been shown that multiple topic markers do not improve our system in comparison to using only the most likely topic, we continued to annotate our source documents with only one topic marker.

5.5 Segmentation of Source Texts

Our first experiments revealed that document size has a noticeable impact on topic models and their perceived sharpness. The initial motivation for segmenting talks into smaller documents was the idea that topics in a talk might change over time. For example, a talk might start out talking about the biology of sea life, and then go on to discuss the creation of life on this planet from a philosophical standpoint. To model the fact that topics can change over time, we segmented our corpus into smaller documents for topic modeling. Manual inspection of the topic models showed that this also resulted in sharper topics, as the context for the topic models had become narrower.

To see the full impact of segmentation on translation performance, we performed multiple experiments with different segmentation sizes both for training the topic models and for inference on the test data. Due to time limit constraints of this work, it was not feasible to evaluate all possible combinations in train and testing document size. We therefore chose configurations we felt closely resembled real world scenarios in which topic modeling might be applied to a speech or lecture translation task.

We evaluated diverging document sizes for training and testing due to practical considerations. It can be assumed that our training data is fully available at training time. In

train docs	test docs	case-sensitive		case-insensitive	
		NIST	BLEU	NIST	BLEU
baseline		6.310	23.05	6.445	24.12
10 lines	1 line	**6.324**	**23.13**	**6.457**	**24.20**
10 lines	windows	6.309	23.09	6.444	24.15
10 lines	10 lines	6.304	23.01	6.438	24.08
windows	windows	6.308	22.97	6.442	24.04
40 lines	40 lines	6.308	23.04	6.441	24.11
talks	1 line	6.297	22.98	6.430	24.05
talks	talks	6.305	23.00	6.440	24.08

Table 5.4: Experiment results regarding differently sized documents used for the training and inference of topics. Document sizes include whole TED talks, chunks of 10 or 40 lines length, sliding windows of 10 lines, and single lines.

an offline translation task, this assumption would also hold true for our test data. In an online task such as simultaneous or near-simultaneous translation of speeches, we would only have the current or the last few lines available to us for topic inference. To model such translation tasks, we always chose a training document size equal to or smaller than the training document size.

The total number of documents we used for training of the topic models varied between 51,022 using sliding windows and 384 using whole talks as documents. Despite these differences, we did not observe an significant impact on training time as training of the topic model only took a few minutes. We expect this is due to our generally small corpus. For all approaches except the sliding window, we consider each word only in one document, so the total number of words used for the topic model stays the same. In the sliding window approach, each word is considered as many times as the size of the sliding window, in our case 10 times. The sliding windows approach might considerably prolong training time on a full-scale system.

The results of our document segmentation experiments are listed in Table 5.4. All configurations were trained on topic models with 100 topics and the most likely topic was added as a topic marker to the data. For all configurations the baseline MERT weights were used as feature weights for the log-linear model. The topic model that was trained on whole talks used the TED-specific stop list as discussed in Section 5.6. All other configurations listed here used the default stop list provided with the MALLET toolkit.

We trained four different topic models: one on whole talks, two on chunks of 40 and 10 lines each respectively and one on sliding windows with a window size of 10 lines. Topics were not trained on single lines because that would defeat the purpose of incorporating context beyond single sentences into our model. Depending on the topic model, topics were inferred on a number of different document sizes for the test data. The topic model trained on 10 line chunks is the same we used in our previous evaluation of the number of topic models, where we reported the result using a single line for topic inference. It is not necessary to retrain a model for inference on differently sized test data documents. The only variation in evaluation scores between configurations that were trained on the same training document size therefore stems directly from the difference in testing document size.

For every topic model, we ran one configuration using the same document size for training of the topic model and topic inference. All of these performed worse than our baseline system. We had improved our baseline result using the topic model trained on 10 line chunks when we inferred topics on single lines, simulating a simultaneous translation task. Using 10 line chunks as documents for topic inference instead, we observe a sharp drop of over 0.1 BLEU points in performance. This is contrary to the intuition that the difference between training and testing data should be as minimal as possible for statistically trained models. We observe this effect when we try to infer topics on single lines from a topic model that has been trained on whole talks. In this case, the performance drops in comparison to the configuration inferring topics on whole talks. Of course the difference between using whole talks and single lines is much larger than the difference in document size when we use 10 line chunks to infer topics on single lines. To further investigate the effect of testing document size, we used a mix between chunks and single lines by inferring topics for our best performing model on sliding windows with a size of 10 lines.

The sliding windows and the single lines approach are the two best-performing configurations. Both were trained on the same topic model trained on 10 line chunks. The top configuration inferred topics on single lines of the test data and is the same we used in our experiments on the number of topics. Using the sliding window approach for inference, we looked at the 4 previous and the 5 following lines of a sentence and used this 10 line chunk to infer topics. We then assigned the most likely topic learned on those lines only to the middle sentence of our 10 line document. This means that in both test sets every sentence may be assigned a different topic. To see how much difference the size of documents has on topic inference, we compared the topic markers of the sliding window approach with the topic markers of the configuration using single lines for inference. Only 16 lines from the test sets were assigned the same most likely topic, which is roughly akin to 1% of the test set. This shows that even a small difference in document size used for inference has a great impact on the inferred topics.

5.6 Stop Lists

In the previous section, we trained and inferred topics on whole talks. As already noted, stop list filtering is important for long documents in topic modeling. The default stop list for German provided with the MALLET toolkit contains only 129 words. When we trained a topic model on whole talks using the default MALLET stop list, we often got one topic consisting mostly of function words belonging to no particular subject. This was fine as long as we inferred topics on single lines, because single lines are so short that they do not contain many function words. The generic topic was not often chosen as the most likely topic for a sentence.

This changes drastically when we try to infer topics on whole talks. In that case, 7 of the 11 talks in the test set were assigned the same generic topic as the most likely topic for the talk. This shows that the topic consisted of words which appear in every talk regardless of its subject. Naturally, a model in which over half of the talks in our test set are assigned the same topic marker cannot help us disambiguate between different contexts. To find a model that would infer different topics for every talk in our set, we had to filter our these function words to avoid a generic topic. We therefore built and evaluated multiple stop lists.

A stop list is a black list that is used to filter out words from our source text prior to topic modeling. We used two corpora to construct stop lists. The first corpus we used was the TED talks data as a task specific corpus, allowing us to filter our the most frequent words appearing in our source texts. As a second corpus we used a larger, general corpus consisting of news articles from the 2012 Workshop on Statistical Machine Translation [CBKM$^+$12], which allowed us to filter out words commonly appearing in German texts.

corpus	# words	# talks assigned the same topic	# different topics
MALLET	129	7	4
news	1000	5	4
news	1500	8	4
TED	1000	8	4
TED	1500	5	7
TED	2000	1	11
news + TED	500	6	5
news + TED	1000	7	5

Table 5.5: Topic inference results on whole talks in the test set using stop lists from the news and TED talks corpora of different length. Given the most likely topic inferred for each talk, shown are the maximum number of talks that have been assigned the same topic and the number of different topics assigned to the 11 talks in the test set. In the stop lists combining both corpora we used the indicated number of words from each corpus.

To build a stop lists, the corpora were lowercased, as our data for topic modeling was also lowercased. A word frequency list was generated consisting of all the words in our corpus and their associated occurrence counts. The frequency list was then sorted according to the occurrence counts and the most frequent words were used for stop word filtering. We also added some hand-picked word groups to every stop list such as single letters, dates and numbers as well as all possible inflections for the most common auxiliary verbs and possessive pronouns.

To evaluate a stop list, we first filtered the preprocessed data using the constructed stop list during the import of the data into MALLET. We then trained a topic model using 100 topics and manually inspected the topics inferred on whole talks of the test set. The results of the inspection are listed in Table 5.5.

From the news corpus we built two stop lists containing the most frequent 1000 and 1500 words respectively. For both configurations, only 4 different topics were assigned as the most likely topic for a talk. Using 1000 words in the stop list, every topic was assigned to at least 2 talks, with the most frequent topic assigned to 5 talks of the 11 talks in our test set. This is a better performance than using the default MALLET stop list, which assigned the same topic to 7 talks, but still does not differentiate between the topics for a talk as much as the topic model needs to. Enlarging the stop list by using the 1500 most frequent words of the news corpus did not help. The resulting stop list inferred the same topic for 8 of our talks. The performance of the topic model does not improve because we are filtering out the wrong words.

The stop lists built for each corpus contain different words. This can be seen when we use the 500 most frequent words from both the TED and the news corpus. The resulting stop lists contains 829 unique entries. This means that of 500 possible words, only about 200 words occurred in both stop lists. Using the 1000 most frequent words from both corpora, the resulting stop list contains 1510 unique words, indicating a word overlap of roughly 450 words. This explains why we do not achieve good results filtering only the frequent words from the news domain. We do not manage to filter out enough stop words occurring frequently in the TED corpus, so they still skew our topic model.

Using a conjunction of both corpora does not help matters much. The resulting topic models still assign the same topic as most likely topic to the majority of talks in our test

configuration	dev	case-sensitive		case-insensitive		MERT
	BLEU	NIST	BLEU	NIST	BLEU	weights
baseline	**23.84**	6.310	**23.05**	6.445	**24.12**	optimized
TED-specific stop list	23.68	**6.316**	23.00	**6.450**	24.07	optimized
TED-specific stop list		6.305	23.00	6.440	24.08	baseweights

Table 5.6: Translation results using a stop list filtering out the 2000 most frequent words from the TED corpus prior to topic modeling. Topics were trained and inferred on whole talks using a model with 100 topics and the talks were annotated with a topic marker for the most likely topic.

set, although we can see a slight improvement in contrast to using only the news domain in our stop lists. In both cases the performance worsens when we use more words from the news set. This might only be a fluctuation in our statistical topic model, but it might also indicate that using the general stop list causes us to filter out words that occur often in news articles, but rather infrequently in our TED talks. This means we filter out words that might actually be helpful in our topic model, deleting valuable information, while we leave in words that occur very frequently in the TED talks and skew our topic model. We therefore abandoned the general news corpus and instead constructed stop lists using the most frequent words only from the TED corpus.

As can be seen in Table 5.5, we achieved good results filtering out only words from the TED corpus. The more words we filtered out, the less talks shared a single topic between them. Filtering out the 1000 most frequent words is not enough, as the resulting topic model still contains a general topic that is inferred for 8 of the 11 test talks. Contrary to the previous stop lists, our results approve when we filter out more words. Using a stop list of 1500 words still infers the same topic for 5 talks of our test set, but a unique topic is inferred for the remaining talks in our set.

This can be further improved by filtering out the 2000 most frequent words of the TED corpus. The resulting topic model infers a different topic for every talk in the test set. The same holds true for topic inference on the development set with this topic model. Manual inspection of the topic model reveals broader topics that sometimes merge words from two subjects, but they are still meaningful topics from a semantic view. To see how well our topic model performed with our DWL, we ran a translation using this model.

Table 5.6 shows the translation results using the previously trained topic model. The topics were trained and inferred on whole TED talks using a topic model with 100 topics. The stop list used for filtering the talks prior to topic modeling contained the 2000 most frequent words from the TED corpus as well as some hand-picked word groups. We added the most likely topic as a topic marker to the unfiltered source texts.

The configuration result using the MERT weights from the baseline was already reported in Section 5.5. As already seen, it performed slightly worse than the baseline. To see the full potential of our approach, we chose to optimize the configuration using MER training with regard to the BLEU score. While it achieves a slightly better NIST score using the optimized weights, we do not manage to improve the BLEU score of our configuration on the test. This might be due to not quite optimal feature weights, as the configuration achieves a low BLEU score on the development set as well.

train docs	test docs	dev	case-sensitive		case-insensitive		MERT
		BLEU	NIST	BLEU	NIST	BLEU	weights
baseline		**23.84**	**6.310**	**23.05**	**6.445**	**24.12**	optimized
windows	windows	23.82	6.303	23.02	6.439	24.10	optimized
windows	windows		6.308	22.97	6.442	24.04	baseweights
baseline		23.84	6.310	23.05	6.445	24.12	optimized
baseline			6.302	23.02	6.435	24.08	topic DWL
10 lines	1 line	**23.88**	6.303	23.01	6.436	24.08	optimized
10 lines	1 line		**6.324**	**23.13**	**6.457**	**24.20**	baseweights

Table 5.7: Experiment results regarding the optimization of MER Training weights.

5.7 MERT Weights

For most of our results we have so far only reported BLEU scores from systems that have been run with log-linear model feature weights that have been inferred using MER training with regard to the BLEU score on the baseline system. We re-used the baseweights to see the effect the inclusion of topics has on the DWL without any variance in the other model parameters. However, the resulting BLEU scores do not represent the true potential of our systems, as different feature weights might allow us to achieve better translation results.

Since we have changed our DWL to include more context, we expect the re-training of our system to assign a larger weight to our DWL, allowing us to gain a stronger influence on translation performance. To examine this hypothesis, we optimized two configurations: the best performing configurations from our experiments on document size (cf. Section 5.5), to see if we could further improve our best result, and the worst performing configuration from these experiments, to ascertain whether the bad performance was due to unfitting feature weights of the baseline. Table 5.7 shows the results from our experiments.

The configuration using sliding windows for both training and inference of the topic model performed the worst in our previous experiments. We therefore optimized the log-linear feature weights using MER training to see if we could improve performance of our system. MER training lifts the BLEU score by 0.05 BLEU points from 22.97 to 23.02. While this is an improvement, the configuration still does not outperform the baseline system. However, the achieved score is now close to the performance of the configuration trained and inferred on documents of 10 lines each. Since the sliding window uses the same context size of 10 lines to infer documents, we would expect the configurations to perform similarly.

We do not see the same improvement when we optimize the configuration that has thus far achieved the highest score. Using MER training to optimize the feature weights for the configuration utilizing topics trained on 10 line chunks and inferred on single lines, we observe a drop in performance of over 0.1 BLEU points on the test set, although our configuration outperformed the baseline on the development set. The optimized weights do not fit our model as well as the baseline weights for the test case. To compare our system with the baseline, we used the weights optimized on the topic DWL to run a new translation with the baseline. The performance of the baseline system diminished as well, although not as sharply at the performance of the topic DWL. Using the topic MERT weights, our baseline now achieves a score 0.01 BLEU points better than the topics DWL. Since the changes in BLEU score between configurations and the baseline are slim, MER training does not always result in optimal BLEU scores on the test set.

5.8 Special Configurations

In addition to investigating the effect of the aforementioned parameters on our topic models, we also evaluated a few special configurations outside the parameter range to gain further insight into the performance of our approach. These special configurations include using keywords instead of topics, training the DWL for a target word only on topics and the source words aligned to the target word and training a DWL only on topic markers without the usual source words of the sentence. In the following sections we will motivate and give a detailed description of each configuration as well as discuss the achieved translation results and their implications.

5.8.1 Keywords

configuration	dev	case-sensitive		case-insensitive		MERT
	BLEU	NIST	BLEU	NIST	BLEU	weights
baseline	**23.84**	6.310	**23.05**	6.445	**24.12**	baseweights
keywords	23.73	**6.316**	23.02	**6.448**	24.09	optimized
keywords		6.303	22.92	6.438	23.99	baseweights

Table 5.8: Experiment results using keywords instead of topics in the DWL.

The TED talks have been annotated by hand with keywords which are available in the meta data for each talk in our corpus. Using these keywords might provide some advantages over trained topic models, as it eliminates the statistical uncertainty of topics. To test this, we ran a configuration using only the keywords associated with each talk as topic markers. No actual topic model was trained.

We extracted the keywords for each talk from the XML meta data provided by the corpus. TED-specific keywords such as *TED Prize*, *TED2009* or *TEDWomen* were removed from the keyword list. All keywords were lowercased. This left us with 246 unique keywords, nearly half of which occur four times or less in our corpus. By far the most used keyword is *technology*, occurring a total of 147 times, followed by *culture*, *global issues* and *science* occurring 115, 107 and 100 times respectively. 36 keywords were used only once. All but one keyword assigned to talks in the development and test sets also occurred in the training set. The number of keywords for each talk varied between 1 and 13. Since the keywords are not ordered in any way, we used all the keywords provided for each talk. To distinguish the keywords from normal words in the text, they were prefaced with a special string. These keyword markers were then inserted in front of every line of a talk, similar to topic markers, and a DWL was trained.

Table 5.8 shows the translation scores using keywords both with the feature weights of the baseline and weights optimized on the keyword-augmented development set. The inclusion of keywords in the DWL worsens the performance of our system in comparison to the baseline. While we do gain 0.1 BLEU points when we optimize the log-linear feature weights using MER training, neither configuration outperforms the baseline system with regard to the BLEU score.

One reason for the low BLEU score lies in the number of keywords used. We investigated the effects using a sentence from a talk by Dan Barber titled "How I fell in love with a fish"[4]. The talk centers around sustainable fish farming and has five keywords

[4] http://www.ted.com/talks/dan_barber_how_i_fell_in_love_with_a_fish.html

source	KEY_agriculture KEY_biology KEY_environment KEY_food KEY_health Sie machten das , indem sie das Land entwässerten .
reference	They did it by draining the land .
hypothesis	They did that , by the entwässerten country .

Figure 5.2: Example of a keyword-augmented source sentence, its English reference and the hypothesis output by our decoder.

assigned. Figure 5.2 shows the keyword-annotated source sentence, the corresponding reference and the hypothesis output by the decoder. The hypothesis generated by the keywords-augmented DWL is the same one as produced by the baseline system. The German word *Land* can be translated as *land* or *country*, depending on the context. In this case, *country* is the incorrect choice, as it refers to a political entity. We would expect a keyword like *agriculture* to favor the translation *land* over *country*, because in the context of farming *Land* will often refer to a field or the soil.

In fact, when we look into the scores produced by the DWL, we find that all keywords, especially the keyword *biology* boosts the DWL score of the target word *country* and discourages the use of *land*, with the exception of one keyword. Unfortunately, the keyword *health* has the opposite influence, favoring the target word *country* and discounting the score for *land*. Table 5.9 shows the DWL scores for *country* and *land* using both the original sentence with all the keywords and the sentence with all keywords except the keyword *health*.

The DWL scores for both words given the original sentence are close together, but excluding the keyword *health* causes the DWL scores to show a marked difference. Although the DWL favors the translation *land* over *country* even with the keyword *health*, the score for *land* is not high enough to cause our decoder to choose *land* as the better hypothesis over *country*. Without the keyword *health*, the score for *land* is higher whereas the score for *country* is lowered. With these adjusted scores, our decoder might have been able to pick the right translation. This shows that even a single keyword can adversely influence our DWL. We observed the same effect using multiple topic markers (cf. Section 5.4). The more topics or keywords are added to a sentence, the higher the chance that one of them cancels out the positive influence of the other keywords or topics.

This example also shows that our DWL does not have enough influence on the final translation score to cause our decoder to correctly disambiguate a homonym. Although the DWL favored the translation of *land* over *country* for our original sentence, the best scoring translation hypothesis output by our decoder contained the word *country* as the translation for *Land*. Even with feature weights of the log-linear model optimized on the keyword DWL, we do not manage to produce *land* as the correct translation for the sentence in Figure 5.2.

target word	all keywords	keywords w/o *health*
country	0.10397	0.04971
land	0.39658	0.63425

Table 5.9: DWL scores for the different target words given the keyword-augmented source sentence and with the keyword *health* removed from the sentence.

5.8.2 Aligned Words

The idea of a word alignment stems from the early word-based machine translation approaches, where the word alignment would define which source words are translated into which target words. To force our DWL to focus on the disambiguation of homonyms, we trained it only on the aligned words and topic markers.

One of the first steps in a statistical machine translation system is the training of an automatic word alignment. Formally speaking the word alignment is a function mapping target words to their corresponding source words in a sentence. Since the word alignment is defined as a function, it allows one-to-many but not many-to-one alignments. To circumvent this problem, we train word alignments in both translation directions and then symmetrize them using a combination heuristic. Nowadays the word alignment is used in phrase-based systems to guide the extraction of phrases.

When we use multiple keywords or topic markers, we lose performance because we get conflicting information sources. By training the DWL for a target word only on the aligned source words and topic markers, we hope to train our model to focus on the relevant words for a translation. For the disambiguation of homonyms, the original source word and the context are important. A source word should be aligned to all its possible translation candidates, and we provide the context through the topic markers. By leaving out the other source words of the sentence, we try to narrow the focus of the DWL on the important features. Relying only on the relevant words and the topic in which the word appeared should give us all the information necessary for the correct disambiguation of a word with few interfering features.

Table 5.10 shows the results for our experiments. We used two approaches to narrow the scope of the DWL. For both approaches we re-used the to date best performing configuration, which employed a topic model with 100 topics that were trained on 10 line chunks and inferred on single lines. All configurations were run using the log-linear model feature weights of the baseline.

In the first approach we took an already trained DWL and filtered the weights output by MegaM after the fact. The DWL is trained with the MegaM toolkit, which outputs a single file for every target word. This file contains every source word that was used as a feature for the training of the target word and its associated weight. In addition to this, the first line contains a bias weight. These weights are then used in the computation of the DWL score as listed in Equation 4.1. We then took the weight file for a target word and filtered out all the weights belonging to the source words that were not aligned to the target word. The bias weight remained in the file. After filtering, we ran translation

DWL features	case-sensitive		case-insensitive		MegaM
	NIST	BLEU	NIST	BLEU	weights
baseline	**6.310**	**23.05**	**6.445**	**24.12**	
aligned words and topics	6.291	22.88	6.425	23.94	trained
aligned words, no topics	6.278	22.85	6.411	23.91	trained
aligned words and topics	6.281	22.84	6.414	23.90	filtered

Table 5.10: Experiment results for DWLs trained only on aligned source words. In two cases the unaligned words were filtered out before training the DWL, once including topics and once without. In an alternative approach the DWL features were filtered after training, but topics remained.

configuration	dev	case-sensitive		case-insensitive		MERT
	BLEU	NIST	BLEU	NIST	BLEU	weights
baseline w/o DWL	23.11	**6.269**	**22.64**	**6.402**	**23.70**	optimized
only topic markers	**23.14**	6.266	22.59	6.400	23.64	optimized
only topic markers		5.997	21.58	6.126	22.59	baseweights

Table 5.11: Experiment results using a DWL trained only on topic markers and not on normal source words. For comparison purposes, the reported baseline system does not include a DWL. The baseweights used as MERT weights for the DWL trained on only topics were taken from the regular baseline system.

using the new weight files as the basis for our DWL score. The configuration achieved a case-sensitive BLEU score of 22.84 points, 0.21 BLEU points less than our baseline.

In the second approach we filtered the source features for each word before training a DWL. For this the MegaM files containing the training features were constructed to only include aligned words and topic markers. The DWL was then trained as usual using the reduced feature set and a translation was run. The configuration performed slightly better than the filtering approach of the trained features, improving the result by 0.04 BLEU points. Filtering out the DWL scores after training retroactively skews the feature weights, and therefore the model cannot perform optimally. We also do not adjust the bias weight to the reduced set. Training a new DWL on the reduced feature set optimizes the weights for the fewer features and therefore improves the performance of the system. However, even re-training the DWL the BLEU results are still 0.17 BLEU points below the scores of our baseline system.

To see how much the topics contributed to our score, we trained a second DWL only on aligned source words without any topic markers. The resulting configuration achieved a BLEU score of 22.85 points, 0.03 BLEU points less than the configuration which included topic markers. Even with the narrow context for the DWL the topics still show a positive influence on our translation results. In order to further evaluate the impact of topics on the DWL itself, we ran a configuration trained only on topic markers without any source words.

5.8.3 Only Topic Markers

Our trained topic DWLs did not improve our BLEU score as much as we had hoped. One reason for this might be the small influence of topics on our overall model. In most configurations, we used the single most probable topic as an additional word in our source sentence. Given an average of 15.6 words per line on the German test set, the influence of a single topic marker on the DWL score might not be enough for our decoder to favor a different translation.

In order to see how much influence topics have on the translation process, we decided to train a DWL using only the topic markers as features. This means the German side of the training corpus for the DWL consisted only of one topic marker per line. No German source words were used as features for the DWL. We then trained the DWL purely on the topic markers. As a result, the topics DWL is a classifier telling us whether or not to use an English source word based solely on the topic of our source sentence.

Table 5.11 shows our evaluation results using a DWL trained only on topic markers. We used the best-performing topic model from our previous experiments, which uses 100 topics trained on 10 lines chunks and inferred topics on single lines of the development and test data. The topic model was not retrained for this experiment. Since the DWL only operates on topic markers and not on normal source words, we do not get the normal positive influence of a DWL on our evaluation score. As a consequence, we compared the DWL trained only on topics to a baseline that does not use a DWL. Apart from the DWL, the baseline system reported here is the same as our normal baseline system. It includes a smoothed 4-gram language model and short-range reordering using lattices. The weights for our modified baseline system were trained using MER training. These weights do not contain a weight for the DWL feature, as it is not present in our baseline. This means we could not use them as baseweights for our DWL trained only on topics for comparison purposes. Instead, we used the weights of the baseline reported in our other results as baseweights.

The configuration using baseweights did not perform well, achieving a BLEU score over 1 BLEU point below the modified baseline system. This is expected, as the DWL trained only on topics is vastly different to a normal DWL. Using MER training on our only topics configuration, we outperformed the baseline on the development set and managed to gain back 1 BLEU point in performance. Unfortunately, the optimized system does not manage to outperform the baseline in the test case. To see the influence our topic model has on the produced hypotheses, we manually looked at the translation results of our model. The results can be found in Section 5.10.

5.9 Additional Evaluation Metrics

In all our experiments we only saw a small increase in BLEU score at best. Arguably, the BLEU score is not the ideal way to accurately measure the improvement of our system with regard to the disambiguation of homonyms. While homonyms do not appear very frequently, translating them incorrectly often has a severe effect on the meaning of a sentence. The BLEU score does not weigh the severity of errors, so a wrongly translated homonym influences the final translation score as much as a wrong article. The only way to truly see if we managed to improve our system performance for a human user would be a manual evaluation. Unfortunately the associated costs and time requirements prohibited us from performing a human evaluation of our system.

To gain further insight into the workings of our configurations without resorting to human evaluation, we investigated two additional automatically calculated metrics. In the first section, we discuss our findings regarding the word accuracy for nouns and verbs of our configurations. Following this, we discuss the Oracle scores achieved in our experiments as an additional performance metric.

5.9.1 Word Accuracy

In this work we employed topic models as a means to include context beyond a single sentence into our decoder in the hope of improving translation performance especially with regard to the disambiguation of homonyms. Even with the best performing configuration we were only able to achieve a slight improvement in BLEU score. The BLEU$_4$ score for a single sentence is calculated as the geometric mean of unigram to 4-gram precision multiplied with a brevity penalty [PRWZ02], which is then averaged over all sentences to gain the BLEU score for a corpus. In this score, all words are weighed the same, meaning there is no difference between correctly translating an article or a noun.

However, it is arguably more important for our system to correctly translate verbs or nouns as opposed to function words, because most of the information of a sentence lies in the verbs and nouns. Since we cannot automatically identify homonyms in our source text, we evaluated the word accuracy of our hypotheses concerning nouns and adjectives as a score that reflects how well we fared translating the meaningful parts of our sentence.

To calculate the word accuracy, we first tagged the preprocessed reference file using the English *TreeTagger* [Sch94], which produces part-of-speech tags using decision trees. The TreeTagger assigned each word a tag from the Penn Treebank tagset [San90]. We then filtered out all the words from the reference that were not tagged as a noun or verb phrase. We did this process only on the reference to minimize the influence of tagging errors on our word accuracy score.

$$\text{WAcc} = \frac{\sum\limits_{(hyp,ref)\in T} |\{\text{words in } hyp\} \cap \{\text{words in } ref\}|}{\sum\limits_{ref\in T} |\{\text{words in } ref\}|} \tag{5.1}$$

Equation 5.1 shows the calculation of the word accuracy. For every hypothesis-reference sentence pair we counted the words occurring in the filtered reference and the hypothesis. This sum of correct words was then divided by the total number of nouns and verbs in the reference. In other words, we calculated the recall for our translation hypotheses, given the filtered reference text.

Table 5.12 shows the word accuracy scores of our configurations. For an easier comparison with the BLEU scores reported in previous sections, we used the configurations run with the baseline feature weights. The filtered reference text contained 11040 words, of which we managed to correctly translate 5700 words at the most using the topic model trained only on topic markers with no source words, and 5666 words at the least using the topic model utilizing keywords instead of topics, as well as the model trained and inferred on chunks of 10 lines each. Our baseline system correctly translated 5691 nouns and verbs of the reference.

configuration	Word Accuracy
baseline	51.55%
only topic markers	**51.63%**
40 lines / 40 lines	51.62%
1000 topics	51.55%
10 lines / single line	51.54%
probabilities	51.54%
talks / talks	51.52%
talks / single lines	51.49%
10 topics	51.47%
aligned words & topics	51.45%
3 topic markers	51.41%
top 5 probabilities	51.37%
10 lines / windows	51.34%
10 lines / 10 lines	51.32%
keywords	51.32%

Table 5.12: Word accuracy scores computed only on nouns and verbs of the reference.

All accuracy scores are within 0.3 percent points of each other, exhibiting less variance than the BLEU scores of our configurations. Far more striking is the fact that the configuration trained only on topics achieved the highest word accuracy score on verbs and nouns, but scored by far the lowest BLEU score out of all configurations we tested. The difference in BLEU score has to stem from words that are not a verb or noun. Our topic models influence many words that are often not critical for the understanding of a sentence, as seen in Example 3 of Figure 5.3. Manually inspecting the differences between our configurations and the baseline show that many changes lie in function word substitutions such as *this* vs. *that*, *to* vs. *of*, *well* vs. *now* at the beginning of a sentence and so on. Choosing one of the other does not change the meaning of the translation, but it does influence the BLEU score. The word accuracy score shows that even using only topic markers for our DWL, we manage to correctly disambiguate more verbs and nouns than our baseline.

The difference in the word accuracy between our best performing configuration according to the BLEU score (10 lines / single line) and the baseline system is a single word. On the other hand the three configurations that have a higher word accuracy than the baseline system achieved low BLEU scores in comparison. This further underlines the difference in these measures. Whenever we manage to translate more verbs and nouns correctly, we lose performance on the other parts of the sentence as evidenced by the BLEU score.

Both the BLEU score and the word accuracy score suffer from the lack of alternative references. Equivalent or even semantically correct translations are counted as wrong whenever they do not appear in our single reference sentence. They are therefore flawed measures for the true accuracy of our system. We do not have alternative references in our corpus as the talks were originally given in English and then manually translated into German. Re-translating the German source text into English is bound to exhibit a greater divergence between the hypotheses and the reference than a normal translation task where we compare our hypotheses to a translation that has been translated as closely to the source text as possible. Machine translation systems by nature translate very close to the source text. As a consequence, the number of perfectly adequate translations that differ from the English reference and are therefore regarded as wrong in the calculation of the BLEU and word accuracy scores may be higher in our system than in a normal system. An example for this is discussed in Section 5.11.

5.9.2 Oracle Scores

The problem of finding the best translation for a sentence is NP-complete. We therefore need a heuristic to efficiently search the space of all possible alternative translations. The KIT decoder uses the beam-search heuristic, which means that at any given point multiple hypotheses are expanded concurrently. If the number of currently active hypotheses exceeds the width of our beam, the lowest-scoring hypotheses are pruned. This process is error-prone, because we risk pruning a hypothesis which might later have achieved the highest score.

At the end of the decoder pass we get a number of alternative translations for the source-sentence ordered by their log-linear scores, the *n-best list*. The top hypothesis from this list is output as the final translation of our system. However, this translation will not always be the best-performing hypothesis with regard to the BLEU score, which is used to measure system performance. Changing the feature weight of the log-linear models does not produce any new translations, but re-scores the existing translations in our system. This has two effects: changing the scores of the hypotheses can change their order in the n-best list, in some cases resulting in a different hypothesis at the top of the list, and it can also change which hypotheses are pruned during the search, therefore including previously disregarded hypotheses in the n-best list.

configuration	BLEU	Oracle
baseline	23.05	32.69
10 lines / single line	**23.13**	**32.74**
keywords	22.92	32.72
talks / single lines	22.98	32.71
10 lines / 10 lines	23.01	32.71
talks / talks	23.00	32.70
10 lines / windows	23.09	32.69
10 topics	23.08	32.69
windows / windows	23.02	32.69
3 topic markers	23.06	32.65
40 lines / 40 lines	23.04	32.65
1000 topics	22.95	32.65
aligned words & topics	22.88	32.59
only topic markers	21.58	32.34

Table 5.13: Experiment results using the BLEU and Oracle scores on the test set.

The *Oracle score* is a measure of how well our system could have possibly performed with regard to the BLEU score. To determine the Oracle score of our system, the BLEU score of all translations in the n-best list is computed and then the best scoring hypothesis is considered to be the translation hypothesis, even though it may not be at the top of the n-best list. Using the best hypotheses for every sentence, the final Oracle score of the corpus is computed. For comparison purposes, all configurations were run using the baseline feature weights.

Table 5.13 shows the BLEU and Oracle scores for our configurations computed on the top 100 hypotheses from the n-best list. The configuration with the highest BLEU score, which employed a topic model trained on 100 topics using 10 line documents and inferred topics on single lines, also achieved the highest Oracle score out of all configurations.

Unlike the word accuracy score, the Oracle score does not measure different things than the BLEU score. We therefore do not see such striking differences in the performance of our systems between the Oracle and the BLEU scores as we did with the word accuracy scores. The Oracle scores are simply a measure for how well our system could have performed given the hypotheses in our decoder. Although few of our configuration manage to outperform the baseline with regard to the BLEU score, it is encouraging to see that many configurations achieve a higher Oracle score than our baseline. This means we do produce better translation hypotheses, but they do not receive a score high enough to push them to the top of the n-best list.

Notably, two configurations using optimized feature weights achieve an even higher Oracle score than the scores reported in Table 5.13. The DWL trained on keywords achieves the highest Oracle score with 33.09 BLEU points, followed by the DWL trained only on topic markers with 32.88 BLEU points. While the inclusion of topics or keywords into the DWL causes our decoder to consider better translation hypotheses than the baseline, the overall influence of topics in the log-linear model is too small for the correct hypotheses to be output as the final translation of our decoder.

5.10 Translation Examples

It is very difficult to pinpoint why a certain configuration performs well or badly in the evaluation. In this section, we report some hand-picked examples that show typical problems in the translation of sentences. We chose these sentences from the translation hypotheses of the DWL trained solely on topic markers, but they are characteristically and occur with all configurations employing topics as source features of the DWL. The DWL trained only on aligned topics achieved the highest word accuracy score of all configurations, and yet performed the worst regarding the BLEU and Oracle scores. To investigate this discrepancy, we took a closer look at the translation hypotheses. The configuration shows the pure influence of topics in the DWL and we found similar examples to those listed here in the output of the other configurations.

Example 1	
source	Wir degradieren einen Menschen , der für unsere Gesellschaft einen Wert hat .
reference	We are effectively grading someone's worth to our community .
baseline	We degradieren a people , the for our society has a value .
only topics	We degradieren a person , the for our society has a value .
Example 2	
source	Deshalb muss man , wenn man von diesen Quellen abhängig ist , einen Weg haben , die Energie auch in Zeiten wenn Sie nicht verfügbar ist , zu bekommen .
reference	And so , if you depend on these sources , you have to have some way of getting the energy during those time periods that it is not available .
baseline	So you have to , if you depending on this sources is a way , the energy in times when you is not available , to get .
only topics	So you have to , if you depending on these wells , have a way of the energy in times when you is not available , to get .
Example 3	
source	Aber sie werden extrem schlimm sein !
reference	But they will be extremely bad .
baseline	But they are going to be very bad .
only topics	But they will be extremely bad !
Example 4	
source	Nun , würden Sie den gleichen Urlaub wählen ?
reference	Now , would you choose the same vacation ?
baseline	Now , you would choose the same holiday ?
only topics	Now , would you choose the same vacations ?

Figure 5.3: Selected translation results showing the difference in the output of the modified baseline without a DWL and the system using a DWL trained only on topic markers.

Figure 5.3 shows the source, reference, baseline and only topics DWL hypotheses for four hand-picked sentences from the test set. The only topics DWL uses the optimized weights from MER training. Each example serves to illustrate a different phenomenon we observed in the hypotheses of our configurations.

In the first example, we managed to improve the baseline translation by choosing *person* instead of *people*. Given the baseline translation one might misinterpret *a people* as *Volk*, meaning an entire nation, when only a single *person* is meant. The ambiguity arises because the German source word *Menschen* can mean both the plural and the accusative of *Mensch* (*human*). Using a topic model we manage to avoid this mistake by preferring *person* as the better translation for *Menschen*.

The second example shows that we do not always manage to improve our baseline, but sometimes even discard a good translation in favor of an incorrect one. The baseline correctly chooses *sources* as the translation for *Quellen*, whereas the DWL trained only on topic markers produces *wells*. Unfortunately, the ten most likely words for the assigned topic give no indication why *wells* should be a more probable translation than *sources* given the topic. While we do not always manage to improve our BLEU score by choosing the words of the reference translation, the cases where we actively worsen our system by choosing a wrong meaning for a word are quite rare.

A very typical example of the differences between the only topics DWL and the baseline system can be seen in example 3. Here, we manage to make no semantic improvements to the sentence, but simply choose different words to say the same thing. The differences in modern English between *will* and *going to* to express a future action are slim to non-existent. In the same way, both *very* and *extremely* express a strong emphasis and can be used interchangeably. In this particular case we manage to improve the BLEU score of our sentence in the only topics DWL by choosing the same words the reference uses. However, for a human user of our system both hypotheses would be perfectly adequate translations of the source sentence. Although we would like to see different nouns or verbs chosen depending on the topic of our source text, we often only get a difference in function words.

Example 4 shows a similar situation to example 3. Even in the cases where we manage to choose a different noun, we often pick a different synonym. In the talk from which the example was picked, we manage to consistently translate *Urlaub* as *vacations* (we do not have the singular *vacation* stored as a translation for *Urlaub* in the phrase table) instead of *holiday* by using the topics DWL. Because of the plural, *vacations* is regarded as wrong as *holiday* by the BLEU score, but a slight boost in BLEU score comes from the re-ordering of *would you*, because it matches higher n-grams. Although we improve the automatic evaluation measures, the meaning of the sentence remains unaffected.

Synonym substitutions between the baseline and the only topics DWL that we observed in our test set include among others *waste* and *trash*, *picture* and *image*, *result* and *outcome*, and *kids* and *children*. In each of these cases, both words are valid translation choices. Whether or not we manage to improve our BLEU score by using one synonym instead of the other depends simply on the luck of matching the reference.

5.11 Poisoned Cookie

Jaime G. Carbonell and Masaru Tomita [CT85] coined the *poisoned cookie* analogy for machine translation. They compared a machine translation system which produces fully accurate, high quality translations 95% of the time, but completely distorts the original meaning and intent of the source text the other 5% of the time, to a cookie jar in which

5% of the cookies are poisoned. Without a way to tell which of the cookies are poisoned, you would not eat a single cookie.

Current machine translation systems produce hypotheses that range from perfectly accurate over slightly disfluent and confusing to downright wrong. Every now and then, we will produce a translation that distorts the meaning of our original sentence, sometimes to a humorous effect. Since these occurrences are usually few and far between, they do not influence our automatic evaluation metrics. Consequently we do not see any improvement in BLEU or NIST score whenever we manage to avoid that pitfall either. In these cases, the evaluation metrics do not accurately portray the improvement of our system, because they can not differentiate between mild and severe errors.

The impact of poisoned cookies on a human user of our system might be rather grave. Choosing the wrong translation for homonyms can often result in mistakes that change the meaning of our sentence. Consider giving a math lecture using an automatic speech recognition and translation system that translates the German word *Vorzeichen* as *omen* instead of *sign*. At the very least, this is an embarrassing and annoying mistake. In the worst case it might keep you from using that system for any future lectures. Avoiding poisoned cookies can therefore be crucial to the success of our system.

source	Fisch Zucht Anlagen verschmutzen die Umwelt , die meisten von ihnen jeden falls , und sie sind ineffizient , nehmen wir Thunfisch .
reference	Fish farms pollute , most of them do anyway , and they are inefficient , take tuna .
baseline	Fish farm sound systems are the environment , most of them , and they are inefficient , let us take Thunfisch .
topic DWL	Fish farm facilities litter the environment , most of them anyway , and they are inefficient , let us take Thunfisch .

Figure 5.4: Example of a *poisoned cookie* translation in our TED corpus. The hypothesis produced by our baseline system wrongly translates *Anlagen* as *sound systems*. We can avoid this wrong translation by using topic models in our DWL.

Figure 5.4 shows an example of a poisoned cookie translation we produce in our baseline system. The German word *Anlage* might be used in the sense of *Musikanlage*, which means *sound system* or *stereo*. In the TED test corpus, the word *Anlage* is used in a talk about fish farming in the sense of a *compound* or *facility*. The original source word *Fischzuchtanlagen* (*fish farms*) was split up during compound splitting into the three words *Fisch* (*fish*), *Zucht* (*breeding*), and *Anlagen* (*facilities*). Our phrase table contains three possible disambiguations of the homonym *Anlagen*: *sound systems*, *facilities* and *plants*, with *sound systems* being by far the most likely translation.

While the preceding word *Zucht* would tell a human translator which type of *Anlagen* is meant, neither the language model nor the DWL of our baseline system are able to choose the correct translation. Instead, they produce *fish farm sound systems*, a nonsensical translation which might confuse or, at best, exhilarate a human reader. In addition to choosing the wrong translation of *Anlagen*, the baseline system omits the translation of the word *verschmutzen* (*pollute*), resulting in a total loss of meaning for the sentence.

Using topics in our DWL, we are able to produce a correct translation in some cases. Due to the statistical nature of topic models, we unfortunately do not produce the correct translation in every configuration employing topic models. One configuration where the positive influence of the topics becomes evident is the system which was trained and inferred on whole talks using the TED-specific stop list (cf. Section 5.6). Talk 790 is assigned a topic about fishing and pollution. Using this topic as a source feature in our DWL, not only are we able to choose *facilities* as the more likely translation for *Anlagen*, but we also translate *verschmutzen* into *litter*. While *(to) litter the environment* might not be a completely correct sentence, it does get the original meaning of our source sentence across. Other configurations that produce the correct hypothesis include all three special configurations (cf. Section 5.8). The DWLs trained on keywords, on only aligned words and on just the topic markers are all able to avoid the poisoned cookie.

Topics can help us produce valid hypotheses and avoid ridiculous mistakes. However, the impact would only be felt by a human user. The new hypothesis of our topic DWL would still score rather low using an automatic evaluation metric such as BLEU or even word accuracy, because they require exact word matches. In the eyes of BLEU, *farm facilities litter the environment* is as wrong as *sound systems are the environment*. Human evaluation would doubtlessly score our topic hypothesis better than the baseline translation, but even a considerable improvement in single sentence would not reflect in the final score over all sentences. To ascertain if we have made a considerable improvement regarding the disambiguation of homonyms, the translations in our corpus would have to be scored and weighed according to the severity of the translation errors. Such an evaluation would be too costly and time consuming to undertake, but the example has shown that topics can help us make a crucial influence on our system beyond the grasp of automatic evaluation metrics.

5.12 German to French Translation Task

The TED corpus is fairly small. One reason why we do not achieve significant improvements using topic models might be the fact that we often do not have the correct translation in our phrase table and are therefore unable to produce it. To see how our approach fares on a larger system, we tested it on a French to German translation task used in the Quaero 2012 evaluation.

The baseline system was trained on Quaero data such as the project Syndicate and Admin, Bookshop and Presseuropa corpora as well as the Europarl corpus, and evaluated on the Quaero P4 test data. The 4-gram language model was trained and adapted on the monolingual sides of the training corpora. In addition to this, a large background language model was trained on parts of the Gigaword corpus and a bilingual language model was trained on bilingual tokens including the target word and all its aligned source words. The baseline system included a POS-based reordering model and a Discriminative Word Lexicon trained on in-domain data. The preprocessing and training of the system was similar to that explained in Section 5.2.

configuration	BLEU	
	case-sensitive	case-insensitive
baseline	37.68	38.47
topic DWL	37.61	38.40

Table 5.14: Experiment results on in-domain data of the Quaero 2012 evaluation German to French translation task using a topic model with 100 topics, trained and inferred on documents of 40 lines length.

The Quaero corpora did not contain any document boundaries. We therefore split the German source sides of the training and test corpora into documents of 40 lines each. We trained a topic model trained on 100 topics and inferred topics on the documents. A topic marker denoting the most likely topic was added to every line of the corresponding document. We then trained a DWL on the topic-augmented in-domain data. The log-linear model feature weights were re-used for the topic configuration.

Table 5.14 shows the BLEU scores for the baseline and topic DWL systems. We do not manage to outperform the baseline system, achieving a score of 0.07 BLEU points less than the baseline. Manual inspection of the translation hypotheses did not reveal any obvious caveats of the topic approach for the Quaero evaluation task. As we have seen on the TED corpus, the correct choice of topic model parameters such as number of topics and document size is crucial for translation quality. The results is very close to the baseline result, indicating that we might improve the result with a different topic model. Unfortunately the time constraints of this thesis did not allow us to optimize topic performance on this translation task by evaluating further configurations.

6. Conclusion

In this chapter we conclude this thesis by summarizing our work and most important findings and recommending directions for possible future work in the field.

6.1 Summary

In this thesis we investigated ways to incorporate context into the decoder of a phrase-based machine translation system to solve the problem of word sense disambiguation. Homonyms are often translated into different words depending on their meaning which can only be discerned through the use of context. Most state-of-the-art machine translation system do not consider any context beyond the single line currently to be translated. Topic models analyze the latent topics inherent in a document. By using topics as additional source features in a Discriminative Word Lexicon, the DWL was able to consider this context information when predicting whether or not a target word should be included in the translation hypothesis. Because the DWL only uses source words as features, it sufficed to train a monolingual topic model. This makes our approach highly scalable for translation tasks requiring translations from one source language into many target languages. The presented approach is also easy to include into existing machine translation frameworks by including the DWL as an additional component in the log-linear model. Furthermore, we avoid data sparsity problems that can occur on small corpora with approaches filtering the trained models instead of utilizing all the training data available.

We first trained a monolingual topic model on the source side of our corpus using Latent Dirichlet Allocation. To denote the most likely inferred topics in our documents, we included topic markers in front of every sentence. The topic-augmented documents were then used to train a DWL, which regarded the topic markers as normal source words. Through this process, they were included as source features in the DWL guiding the translation decisions of our decoder. To evaluate our model, we translated the TED corpus consisting of public educational talks on a variety of subjects from German to English.

There are many parameters influencing the quality of the topic model that have to be considered, such as the number of topics used in the model, the document size on which topics are trained and inferred and the number of topic markers to be included in the documents. We undertook an extensive study of these parameters in order to evaluate their influence on the translation performance. In early experiments, 100 topics was found to be a reasonable number of topics to use and was subsequently held constant for all other experiments. We also found that using a single topic marker resulted in higher BLEU scores than including multiple topic markers or probabilities. The segmentation of our corpus into smaller documents for topic modeling was an integral part of our work. We found that smaller chunks generally worked better than using whole talks as documents. Using Minimum Error Rate Training to optimize the feature weights of the log-linear model did not necessarily result in higher BLEU scores on the test set, as the improvements in BLEU score of our models were generally slim to nonexistent.

In addition to the study of parameters, we evaluated DWLs trained on keywords instead of topics, trained only on aligned words and topic markers, and trained only on topic markers. Although they did not outperform our baseline system in terms of BLEU score, we gained valuable insight into the workings of our models.

Since the BLEU score is not ideally suited to measure translation improvement regarding the disambiguation of homonyms specifically, we reviewed the word accuracy computed on nouns and verbs as well as the Oracle scores of our configurations. Translation examples further underlined that while the improvements in BLEU score were slim, we did manage to improve the translations of some homonyms. However, the largest difference between the hypotheses and the baseline translation was found in punctuation, local word reorderings and substitutions of valid translation alternatives and function words. Our approach might still prove of value to a human user, because we were able to restore the meaning of sentences that was lost in the translations of the baseline system.

6.2 Outlook

While we were not able to achieve significant improvements in the translation results through the incorporation of topics into a Discriminative Word Lexicon, we did gain valuable insight into the workings of our approach. Following the results of our experiments, several interesting lines of research might be pursued in the future.

Our experiments have shown that topics positively influenced the translation decisions of our decoder. However, as evidenced by the Oracle Scores and the DWL trained only on topics, our current approach does not factor in the topics strongly enough. One might therefore try to significantly increase the weight of topic markers over source words in the training of the classifiers or the calculation of the DWL score. Another approach would be to train two separate DWLs, one trained only on topic markers and one trained on the regular source words of the corpus. This would allow the weighing of the topic DWL through Minimum Error Rate Training while preserving the positive influence of a regular DWL on translation results.

The topic models did not only facilitate the disambiguation of homonyms, but also caused our decoder to frequently choose different function words than the baseline system. To reduce the influence of topics on words whose translation is not topic-dependent, one could train the most frequent words appearing in the corpus as a normal DWL without any topics. Because these words appear frequently, we can assume to have good translations for them stored in the phrase table and the language model should have significant n-gram counts to choose the correct word. By including topics only in the DWLs of the less frequent words, we use the context only when our models have few training occurrences and therefore unreliable estimates of their translation frequencies. The optimum number of high-frequency words excluded from topic modeling would have to be determined through experiments.

We evaluated our experiments on a comparatively small corpus. In some cases we observed we did not have the correct translation alternative stored in our phrase table and were therefore unable to produce it even with the incorporation of topic models in our system. A larger corpus would alleviate this problem. Without the concern of data sparsity, another approach to consider would be the training of topic-specific DWLs. Instead of explicitly including the topics as source features in the DWL, one might use them implicitly by training multiple DWLs, each on the parallel data pertaining to a specific topic. Subsequently, instead of using the same DWL for all sentences, the DWL matching the topic of the source sentence would be used to score the translation hypotheses. This approach has been used successfully in the past with topic-adapted translation lexicons and filtering of phrase pairs.

In the end, only an evaluation of our approach through humans could accurately measure the usability of our system, because unlike automatic evaluation metrics humans are able to weigh the severity of translation errors and recognize valid translation alternatives. Unfortunately, to undertake such a study would have exceeded the scope of this thesis. It is our belief that a human evaluation would reflect the improvements in disambiguation we achieved with our approach.

Bibliography

[Ble12] D. M. Blei, "Probabilistic Topic Models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, April 2012.

[BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, March 2003.

[BPPM93] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics - Special issue on using large corpora: II*, vol. 19, no. 2, pp. 263–311, June 1993.

[CBKM$^+$12] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 workshop on statistical machine translation," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Montréal, Canada, June 2012, pp. 10–51.

[CGF12] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT 2012)*, Trento, Italy, May 2012, pp. 261–268.

[CT85] J. G. Carbonell and M. Tomita, "New Approaches to Machine Translation," in *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages (TMI 1985).* Hamilton, New York: Association for Computational Linguistics, August 1985, pp. 59–74.

[Dau04] H. Daumé III, "Notes on CG and LM-BFGS Optimization of Logistic Regression," August 2004, paper available at http://pub.hal3.name#daume04cg-bfgs, implementation available at http://hal3.name/megam/.

[Dod02] G. Doddington, "Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics," in *Proceedings of the Second International Conference on Human Language Technology Research (HLT 2002).* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 138–145.

[Fed02] M. Federico, "Language Model Adaptation through Topic Decomposition and MDI Estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, vol. 1, May 2002, pp. I–773 –I–776.

[GH99] D. Gildea and T. Hofmann, "Topic-Based Language Models Using EM," in *In Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH 1999)*, Budapest, Hungary, September 1999, pp. 2167–2170.

[GZZ10] Z. Gong, Y. Zhang, and G. Zhou, "Statistical Machine Translation Based on LDA," in *4th International Universal Communication Symposium (IUCS 2010)*, October 2010, pp. 286 – 290.

[HG06] B.-J. P. Hsu and J. Glass, "Style & Topic Language Model Adaptation Using HMM-LDA," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 373–381.

[KHB+07] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*. Prague, Czech Republic: The Association for Computational Linguistics, June 2007, pp. 177–180.

[Koe05] P. Koehn, "Europarl: a Parallel Corpus for Statistical Machine Translation," in *Conference Proceedings: the Tenth Machine Translation Summit (MT Summit 2005)*. Phuket, Thailand: AAMT, September 2005, pp. 79–86.

[Koe10] P. Koehn, *Statistical Machine Translation*, 1st ed. Cambridge University Press, 2010.

[McC02] A. K. McCallum, "MALLET: a Machine Learning for Language Toolkit," 2002, http://mallet.cs.umass.edu.

[MCN+11] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French Translation Systems for IWSLT 2011," in *International Workshop on Spoken Language Translation*, San Francisco, California, USA, December 2011, pp. 73–78.

[MHN09] A. Mauser, S. Hasan, and H. Ney, "Extending Statistical Machine Translation with Discriminative and Trigger-based Lexicon Models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 (EMNLP 2009)*. Association for Computational Linguistics, 2009, pp. 210–218.

[Och03] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (ACL 2003)*. Association for Computational Linguistics, 2003, pp. 160–167.

[ON03] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[PRWZ02] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318.

[RF11] N. Ruiz and M. Federico, "Topic Adaptation for Lecture Translation through Bilingual Latent Semantic Models," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 294–302.

[RV07] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-based Distortion Model," in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*, Skövde, Sweden, September 2007, pp. 171–180.

[San90] B. Santorini, "Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision)," University of Pennsylvania Department of Computer and Information Science, Philadelphia, PA 19104, Tech. Rep. Technical Report No. MS-CIS-90-47, July 1990.

[Sch94] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *Proceedings of the International Conference on New Methods in Language Processing (NeMLaP 1994)*, Manchester, UK, September 1994.

[Sto02] A. Stolcke, "SRILM – an Extensible Language Modeling Toolkit," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, Colorado, USA, 2002, pp. 901–904.

[TJBB06] Y. W. The, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[TLS07] Y.-C. Tam, I. Lane, and T. Schultz, "Bilingual LSA-based Adaptation for Statistical Machine Translation," *Machine Translation*, vol. 21, pp. 187–207, December 2007.

[Vog03] S. Vogel, "SMT Decoder Dissected: Word Reordering," in *Proceedings of the 2003 International Conference on Natural Language Processing and Knowledge Engineering (NLPKE 2003)*, Beijing, China, October 2003, pp. 561–566.

[XXZ$^+$12] X. Xiao, D. Xiong, M. Zhang, Q. Liu, and S. Lin, "A Topic Similarity Model for Hierarchical Phrase-based Translation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2012)*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 750–758.

[ZX08] B. Zhao and E. P. Xing, "HM-BiTAM: Bilingual Topic Exploration, Word Alignment, and Translation," in *Advances in Neural Information Processing Systems 20 (NIPS 2007)*. Cambridge, MA: MIT Press, 2008, pp. 1689–1696.