Sequence Segmentation for Statistical Machine Translation

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades einer Doktorin der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatikerin Jia Xu

aus

Shanxi

Berichter: Universitätsprofessor Dr.-Ing. Hermann Ney
Universitätsprofessor Dr. Dekai Wu

Tag der mündlichen Prüfung: 10. September 2010

Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.

– Albert Einstein, 1929

Acknowledgments

Appreciation is the best motivation. During my Ph.D. study, I received invaluable advice and care. Here I would like to express my gratitude to people who supported and accompanied me along my work.

First, I am deeply indebted to my advisor, Prof. Dr. Hermann Ney, for his constant support. Without his help, this work would not be possible. He gave me the opportunity to attend a variety of conferences and the possibility to work for international projects. The essence of his front and academic attainments and rigorous diligent research style deeply affect my future work. Teacher graciousness is unforgettable!

I am very grateful to Prof. Dr. Dekai Wu for agreeing to take time to evaluate this thesis as a co-referee and for attending my defense in Germany. I would also like to thank the members of my committee: Prof. Dr. Matthias Jarke and Prof. Dr. Joïgen Giesl. Their advice and attendance are appreciated.

I am sincerely grateful to Dr. Jianfeng Gao for his scientific guidance and support all these years. I am also thankful to Dr. Kristina Toutanova, Dr. Yuqing Gao, Dr. Yonggang Deng and other researchers at Microsoft and IBM Research for their helps during my Internships in the United States. Special thanks go to Dr. Werner Hemmert and Prof. Dr. Klaus Obermayer who supervised my Diploma thesis and led me into the area of research.

My Ph.D. study turned out to be an unforgettable experience in Aachen, mostly thanks to the support from my colleagues: Richard Zens, Sharahm Khadivi, Evgeny Matusov, Christoph Schmidt, Arne Mauser, Yuqi Zhang, Jessica Kikum, Jan Bungeroth, Gregor Leusch, Saša Hasan, David Vilar, Björn Hoffmeister, Daniel Keysers and all other individuals. I would like to greatly thank colleagues who proofread this thesis. I also thank the machine operators and secretaries of Informatik 6.

At this point, I would like to express my everlasting gratitude to my dearest family: my father Xihua Xu and my mother Yukun Zhang. Their deepest love, encouragement, patience, support and education have been accompanied and encouraged me all the time and are the most precious wealth in my life.

This dissertation is dedicated to all care, help, support and encouragement of my relatives, teachers, colleagues, students and friends!

Abstract

In the last decade, while statistical machine translation has advanced significantly, there is still much room for further improvements relating to many natural language processing tasks such as word segmentation, word alignment and parsing. Human language is composed of sequences of meaningful units.

These sequences can be words, phrases, sentences or even articles serving as basic elements in communication and components for computational modeling. However, in monolingual text some sequences are not naturally separated by delimiters, and in bilingual text both sequence boundaries and their corresponding translations can be unlabeled. This work addresses solutions of sequence segmentation and alignment for statistical machine translation, including the following topics:

Chinese word segmentation: Different from the explicit word segmentation in trivial approaches, I introduce integrated Chinese word segmentation, where segmentation and alignment of words are trained jointly, and the decoding is performed on the lattice composed of alternative word segmentations. I show that direct translation on Chinese characters can achieve even better translation performance than translation on Chinese words;

Phrase training: Currently phrases are extracted in a heuristic way. I propose a mixture phrase pair model which is trained discriminatively allowing to combine multiple extraction processes and various resources, especially the underlying word alignment models discarded in the standard approach;

Parallel sentence exploitation: Training corpus acquisition is crucial for a data-driven translation system. I propose a maximum-entropy model where document pairs are partitioned recursively into sentence pairs using 'binary segmentation' without any requirement on sentence boundary markers;

Domain adaptation: A hierarchical clustering algorithm is applied to classify the training data into distinct domains. Domain specific language models and translation models are then combined to build a domain dependent system, and domain priors are estimated with a minimum error rate training.

Experimental results on state-of-the-art, large-scale Chinese-English tasks show that the training speed can be increased with a factor of four and each above mentioned method leads to an enhancement of the translation quality up to 6% relatively.

Zusammenfassung

Menschliche Sprache besteht aus Sequenzen sinnvoller sprachlicher Einheiten. Diese Einheiten können Wörter, Phrasen, Sätze oder Artikel sein, die als Basiselemente in der Kommunikation und als Komponenten für die maschinelle Modellierung dienen. Aller-dings sind die Definitionen von einigen Sequenzen wie der von Phrasen und chinesischer Wörter nicht eindeutig, da keine Trennsymbole im Text existiert. Dies stellt eine Anforderung an viele Sprachverarbeitungsaufgaben dar, zum Beispiel in der maschinellen Übersetzung. Wenn ein Text automatisch von einer Sprache in eine andere Sprache übersetzt wird, kommen die Sequenzen paarweise in beiden Sprachen vor. Eine wesentliche Aufgabe ist die Erkennung der Sequenzen in der Quellsprache und deren entsprechenden Übersetzungen.

Diese Arbeit stellt Lösungen der Probleme der einsprachigen und zweisprachigen Sequenzsegmentierung für die statistische maschinelle Übersetzung vor, die sich auf die Segmentierung und Alignierung von Wörtern, Phrasen, Sätzen und Dokumenten beziehen. Wörter im chinesischen Text sind nicht durch Separatoren getrennt, was die chinesische Sprache von den meisten europäischen Sprachen unterscheidet. Ein allgemein verwendeter Ansatz in der Chinesisch-Englischen Übersetzung ist die Verwendung von expliziter Wortsegmentierung, indem die chinesischen Wörter erst segmentiert und dann mit dem Standardverfahren übersetzt werden. Diese Art der Wortsegmentierung ist nicht notwendigerweise optimal für die Übersetzung. Wir setzen eine halb-überwachte Wortsegmentierung ein, die einsprachige und zweisprachige Informationen berücksichtigt, um eine geeignete Segmentierung für die Übersetzung abzuleiten. Die Alignierung und Segmentierung von Wörtern werden durch das sogenannte „Gibbs Sampling" gleichzeitig trainiert. Neue Wörter werden nach dem Prinzip des Bayes'schen Lernen generiert. Darüber hinaus werden unterschiedliche Wortsegmentierungen in einem Wortgraph repräsentiert und bei der Suche nach der besten Übersetzung berücksichtigt. Die Segmentierungsentscheidung ist auf diese Weise in die Dekodierung integriert.

Die Phrasenpaare, die als Sequenzen von Wörtern und deren Übersetzungen definiert werden, bilden ein weiteres Kernelement im Aufbau des Übersetzungssystems. Im Standardverfahren sind die Phrasenpaare heuristisch extrahiert basierend auf der besten Wort-alignierung, während die zugrunde liegende Wortalignierungsmodelle verworfen werden. Um diese Information einzubeziehen, führen wir ein Mixture-Modell ein, das unterschiedliche Modellableitungen kombiniert. Verschiedene Extrahierungsprozesse und Ressourcen können zur Generierung der Phrasenpaare beitragen.

Parallele Sätze und domänspezifische Korpora, die im Training verwendet werden, sind für die Leistung des datengetriebenen Übersetzungssystems von entscheidener Bedeutung. Wir werden daher einen neuartigen Ansatz vorstellen, mit dem wir die satzalignierten Daten erhalten, indem wir die zweisprachigen Dokumente rekursiv in zwei Teile aufteilen. Diese Methode übertrifft die Leistung der allgemeinen Satzalignierungsmethoden und setzt keine Ankerwörter an den Satzgrenzen voraus, was besonders interessant für Transkriptionstext ist. Darüber hinaus führt die Verkürzung von langen Satz-

paaren zu einem effizienteren Training und zu einer höheren Qualität in der Wortalignierung. Da immer grössere Menge an Trainingsdaten einbezogen werden, gibt es einen grösseren Bedarf an Domañenanpassungen. Wir diskutieren Clusteralgorithmen, um domainabhängige Sprachmodelle und Übersetzungssysteme aufzubauen. Die vorgeschlagene Methode fordert viel weniger zweisprachige Daten als normalerweise für den Aufbau eines domainabhängigen Systems verwendet werden. Dieses Verfahren ist einfach und effizient, um viele Domänen zu erfassen. Abschliessend werden wir die Ergebnisse der Experimente der oben genannten Methoden vorstellen. Die Qualitaẗ im Bereich der Chinesisch-Englischen Übersetzungsaufgaben ist gegenüber dem Stand der Technik signifikant verbessert worden und das Modelltraining ist effizienter. Weiterhin werden wir zeigen, dass es ein besseres Übersetzungsmodell gibt, das direkt die chinesische Zeichen statt der chinesischen Wörtern übersetzt.

Contents

1 Introduction **3**

2 Scientific goals **5**

3 State of the art **6**

3.1 Chinese word segmentation 7

3.2 Sentence segmentation 10

3.3 Phrase pair segmentation 11

3.4 Document segmentation 12

4 Chinese word segmentation **15**

4.1 Problem description 15

4.2 Definition of Chinese word segmentation 15

4.3 Common methods 16

4.4 n-gram segmentation 17

4.5 Segmentation learned from the alignment 21

4.6 Semi-supervised Chinese word segmentation 23

4.6.1 Approaches 23

4.6.2 Generative model 25

4.6.3 Final model 28

4.6.4 Gibbs sampling training 29

4.6.5 Computing probabilities of alternatives 30

4.6.6 Determining the set of alternative hypotheses 34

4.6.7 Complete segmentation algorithm 35

4.7 Integrated Chinese word segmentation in search 37

4.7.1 Integrated Chinese word segmentation model 38

4.7.2 Constructing segmentation lattices 38

4.7.3 Weighting segmentation lattices 42

5 Phrase pair segmentation 44

5.1 A mixture phrase model 45

5.2 Phrase model features 47

5.2.1 IBM model 1 based on word posterior probabilities 47

5.2.2 HMM based on word posterior probabilities 48

5.2.3 Bilingual entropy 49

5.3 Discriminative training 49

6 Sentence segmentation 52

6.1 Binary segmentation 52

6.2 Segmentation model 54

6.2.1 Normalized IBM model 1 55

6.2.2 Other features and alignment concatenation 55

6.2.3 Efficient IBM model 1 computation 57

6.2.4 Segmentation example 58

6.3 Bitext exploitation 59

7 Document segmentation 61

7.1 Document clustering 61

7.2 Building domain specific translation systems 63

7.3 Extremum of functions 65

7.4 Domain adaptation 66

7.4.1 Language model based domain identification 67

7.4.2 Information retrieval approach 67

8 Results 68

8.1 Evaluation criteria 68

8.2 Task and corpus statistics 69

8.3 Chinese word segmentation 77

8.3.1 Statistics of the word length in the dictionary 77

8.3.2 Translation results 78

8.3.3 Analysis of translation outputs 81

8.3.4 Conclusions 82

8.4 Phrase pair segmentation 84
8.4.1 Translation results 84
8.4.2 Analysis on translation outputs 85
8.4.3 Conclusions 86
8.5 Sentence segmentation 87
8.5.1 Segmentation parameters 88
8.5.2 Translation results 88
8.5.3 Conclusions 90
8.6 Document segmentation 92
8.6.1 Classification results 92
8.6.2 Translation results 93
8.6.3 Conclusions 94

9 Scientific contributions **96**

10 Conclusion **99**

Chapter 1

Introduction

The goal of this thesis is to show the impact of sequence segmentation on the performance of a state-of-the-art statistical machine translation system.

In mathematics a *sequence* is defined as an ordered list of elements, in which the same elements can appear multiple times at different positions. In natural language processing (NLP) a character is the smallest unit in a text. A sequence of characters forms a word, and a sequence of words forms a phrase or a sentence. *Segmentation* is a process to divide a sequence into meaningful sub-sequences, which includes detecting word boundaries, extracting phrase pairs, chunking texts as well as clustering topics.

Statistical machine translation (SMT) is a sub-field of NLP and addresses the problem of automatically translating a text of one language into a text of another language using machine learning techniques and statistical modeling approaches. Identification of translation model units is therefore an elementary issue for SMT and may result in a bottleneck in developing a high-performance translation system.

While most of the European languages have explicit word boundary markers, in Chinese texts, words are not separated by delimiters. This leads to the *Chinese word segmentation (CWS)* problem. A widely used approach is to apply a Chinese word segmenter trained from manually annotated data using a fixed lexicon. Such word segmentations are not necessarily optimal for translation. We propose a semi-supervised Chinese word segmentation model which uses both monolingual and bilingual information to derive a segmentation suitable for SMT. On a test corpus we take all possible segmentations of a sentence into account and integrate the segmentation decision into the search for the best translation. This allows the Chinese word segmentation model to be estimated for the use in SMT and leads to improvements in translation performance.

Once the words are defined, the translation is performed based on phrases instead of on single words to capture context dependency. Therefore, generating the proper phrases and their corresponding translations is a major issue in the translation task, called *phrase pair segmentation*.

Usually, the phrase pairs are extracted heuristically based on the word alignment results, and the translations are performed afterwards. We will present a discriminative phrase pair training algorithm which is parameterized with feature functions; the weights of these

features are then optimized directly with respect to the end-to-end system performance. Multiple data-driven feature functions are proposed to create a balance between precision and recall.

With refined translation models in general, we meet higher demands on parallel sentences to train large-scale data-driven SMT systems. But plenty of bilingual resources are only available on the document level. We will show novel methods to align sentences to their translations by partitioning document or paragraph pairs which is referred to as *sentence segmentation*. This approach can also effectively reduce the computation requirement in word alignment training without deterioration of the translation performance.

While statistical machine translation has advanced significantly with better modeling techniques and much more training data, domain specific SMT has received much less attention and leaves much room for further improvements. In this work, we address domain issues and use a combination of feature weights and language model adaptation in order to distinguish multiple domains, which share a general translation engine with phrase-based log-linear models. The proposed method requires much less parallel data than what is typically used to build a domain independent system, which makes it easy, cost-effective and efficient to capture as many domains as required. Domain adaptation during decoding is approached by source text classification methods. The results of proposed method show significant improvements of the proposed domain dependent translation over domain independent translation.

Throughout this work, in order to allow a meaningful comparison of different segmentation approaches, the proposed methods are applied to two well-known translation tasks in a tourism-related domain for small data track and in a news domain for large data track. Results of experiments demonstrate consistent and significant improvements on translation performance over the widely used standard sequence segmentation approaches. The main contributions of this work are summarized on Page **??**.

Structure of this document

The work is organized as follows: Chapter **??** describes the advanced methods of CWS in machine translation, which includes Bayesian semi-supervised CWS in the training of the word alignment and the integrated CWS into the search for the best translation. Chapter **??** discusses the mixture phrase segmentation method that combines features derived from underlying word alignment models instead of from single-best word alignments. In Chapter **??** we employ sentence segmentation for efficient training and data exploitation, and in Chapter **??** we address the domain issue and present an unsupervised clustering method for language model adaptation. Finally, in Chapter **??** we present the corpora used in the experiments, summarize the methods used in current research and discuss the error rates presented by other researchers.

Chapter 2

Scientific goals

The goals set out at the starting point of the work for this thesis (and supplemented at different points in time along the work) were to

- improve the state-of-the-art translation performance for machine translation systems (measured by automatic evaluation criteria) using enhanced statistical sequence segmentation models;
- investigate Bayesian methods used in large scale, practical NLP tasks;
- integrate Chinese word segmentation into the application of statistical machine translation;
- update the heuristic phrase extraction method with a sound phrase training model;
- explore automatically bitext for data-driven MT;
- train system efficiently with reduced computational requirements;
- translate text depending on its domain.

Chapter 3

State of the art

This chapter provides a brief overview of the state-of-the-art methods related to sequence segmentation for machine translation (MT). These methods serve as baselines for a comparison and as a starting point for later sections. We introduce the methodologies of statistical machine translation and address major problems to be solved along this thesis. We also anticipate results based on proposed solutions discussed in later chapters and put them into their frame of reference here.

Machine translation is a sub-field of computational linguistics that investigates the use of computer software to translate text or speech from one natural language into another. The history of machine translation begins in the 1950s. There are mainly three approaches which address the problem of machine translation: Rule-based methods parse a text, usually creating an intermediary, symbolic representation, from which the text in the target language is generated. Statistical machine translation generates translations using statistical methods based on bilingual text corpora, a corpus in a source language together with its translation corpus in a target language. The example-based machine translation (EBMT) is often characterized by its use of a bilingual corpus as its main knowledge base at run-time, which is essentially a translation by analogy and can be viewed as an implementation of a case-based reasoning approach of machine learning.

The currently most widespread and top-performing approach is the statistical machine translation. Therefore, in this work we will only focus on methods and methodologies in the field of SMT. The basic work flow of a statistical machine translation is composed of data exploitation, preprocessing, training, translation and evaluation. Sequence segmentation plays an important role along these steps.

With the increasing availability of online corpora, data-driven approaches have become central to the natural language processing community. Statistical machine translation is a data-driven process. Translation systems are trained based on the assumption of a set of parallel sentences. Automatical data exploitation and employment is in high demand and seen as a new challenge for MT. We will describe the baseline methods for **sentence segmentation and alignment** in Section **??**, the novel methods in Chapter **??** and show experiments in Section **??**.

The bilingual raw texts obtained from various resources need to be prepared and converted into a machine-readable format. For example, punctuations are tokenized from English

words. Ambiguities in words are truncated and sentence pairs which are too long are filtered out or shortened. These steps apply to all language resources. In addition, some special handling on certain languages is necessary. One of the most crucial problems is **Chinese word segmentation**, because Chinese word boundaries are not naturally given but have to be detected. We will discuss the baseline methods for this problem in Section **??**, the improved methods in Chapter **??** and experimental results in Section **??**.

In statistical approaches we take pre-defined models and train model parameters based on labeled data. In MT, the input for training is the sentence aligned corpus. Given a source language sentence, which is to be translated into a target language sentence, the model is designed to find the most probable translation among all possibilities. A standard training produces the **word alignment** output, which indicates one or more words in the source sentence are to be translated into one or more words in the target sentence. In Chapter **??**, we will present a new training method that jointly aligns and segments words and which leads to and enhanced translation quality.

Based on the model and trained parameters, best translations are generated. To capture the context information and local reordering, word groups, so-called phrases are extracted to form the translation units. The search for the best translation is performed on a set of phrase pair sequences. The target side of the selected phrase pairs compose a sentence. The conventional **phrase pair generation** is only based on the word alignment input, which is heuristic and not very accurate as to be described in Section **??**. In Chapter **??** we will show that phrase pairs induced from various data resources using a combination of multiple features results in a significant improvement in translation performance. Experiments will be presented in Section **??**.

Finally, text styles and topic contents need to be considered for a refined translation and a sound formulation. **Domain adaptation** is therefore investigated to automatically identify user-specific domains with this aim in mind. The methods introduced by other authors are stated in Section **??**, and we propose language model adaptation in Chapter **??**, which is shown to be very effective to improve translation quality and which is widely applied by other researchers, along with the results in Section **??**.

Now, we will briefly overview the previous work on these problems in the following sections.

3.1 Chinese word segmentation

In contrast to most of the European languages, words in Chinese texts are not separated from each other, which is difficult and poses an essential problem for many natural language processing tasks. Detecting Chinese word boundaries automatically is therefore widely applied in areas such as text processing, Chinese character input, speech recognition and information retrieval.

A word is the smallest meaningful unit of a language. However, a clear definition of a Chinese word does not exist so far but depends on the application. There are two major problems in CWS: ambiguities and words not contained in the lexicon. An ambiguity occurs if a given Chinese sequence has more than one manual word segmentation. For instance, '不满意(not satisfied)' can be segmented into '不(not) 满意(satisfied)' or '不

满(not satisfied) 意(meaning)' with different meanings, and '马上(immediately)' can either be treated as one word or separately as '马(horse) 上(on)'. This problem has troubled linguists for decades. In some cases, even a human can hardly decide which segmentation is correct. The other problem concerning words out of lexicon is that some words exist but are not collected in the manual predefined dictionary. Named entities and professional glossaries are typical examples of words that are difficult to capture in advance. For instance, a corpus of legal language with 15K entries may contain 30% out-of-vocabulary words given a manual dictionary with 70K word entries.

The performance of Chinese word segmentation is usually evaluated by three criteria: precision, recall and F-measure. Precision is defined as the number of correctly segmented words divided by the number of all words after segmentation. Recall is defined as the number of correctly segmented words divided by the number of words in the reference document. And F-measure is calculated based on precision and recall. These criteria can measure the quality of Chinese word segmentation given a reference with correct segmentations. However, a real standard segmentation does not exist but depends on the application and context. In the experiments we observed that the correlation of word segmentation quality and translation error rates are not high enough. Therefore, we have abandoned the above measures and take only the translation error rates as the final evaluation criteria to measure the Chinese word segmentation performance.

In Chinese-to-English machine translation, the common approach has been to segment the Chinese text using an off-the-shelf CWS method and to perform a standard training and translation process once the segmentation is fixed. There are many ways to recognize word boundaries. The simplest method is to use the **maximum matching**. Characters in a sequence are checked whether they match the words in the dictionary from left to right; first the longest words are checked, then shorter words. An **inverse maximum matching** matches words from right to left. This method is rather naive because words with equal lengths are treated in the same way. Statistical methods are introduced to estimate model parameters based on the training data. The advantages of statistical approaches are robustness and generalization. By assigning a probability to each word entry, high frequency and low frequency words are distinguished. Under various statistical methods, the **N-gram** based Chinese word segmentation is widely applied, such as in [**?**], [**?**], [**?**]. We will present this method in detail in Section **??** and use the unigram word segmentation as one of the baselines in the experiments, which requires a manual lexicon containing a list of Chinese words and their frequencies. The lexicon and frequencies are obtained using manually annotated data. **HMM** is another statistical model to perform word segmentation. In [**?**], named entities are taken as features besides the segmentation lexicon. Bigram class dependency and word conditional probability are taken into account in the HMM based framework to search for optimal segmentation boundaries. **Maximum entropy** [**?**] is a similar approach that combines user defined features and segmentation decisions. Single character words, characters at the beginning, in the middle and at the end of a word constitute four basic classes of the model. [**?**] employed a conditional random field (**CRF**) model for sequence segmentation.

The increasing amount of monolingual data encourages researchers to generate word lexica automatically, which may be more domain-specific. [**?**] developed a Chinese word segmenter by only using a manually segmented corpus, where segmentation rules were

extracted from the corpus. [?] and [?] used neither a dictionary nor a segmented corpus. The input texts are grouped into pairs which have the highest value of mutual information. This can be learned from the Chinese monolingual corpus. [?] firstly introduced t-tests to measure word correspondences. Recent investigations were carried out taking multiple features into account. [?]'s models included morpho-syntactic information and adapted CWS to domain specific environments.

Nonetheless, those methods are still not necessarily optimal for translation for the following two reasons: 1. The segmentations may be erroneous, because the context varies. 2. The best segmentation for a given character sequence depends also on its translation. For a given character sequence the 'correct' segmentation is not universal, but we need to consider the context and the language to be translated into. So far, a Chinese word segmentation method with dominant advantages has not been found. Since most methods are not specifically developed for the MT application, significant improvements in translation performance have not yet been shown to result from using these more sophisticated methods for CWS. Therefore, designed with the translation task in mind, we investigate a joint word segmentation and alignment approach with integrated Chinese word segmentation in the search for the best translation.

The main characteristic of the proposed Chinese word segmentation method is that the segmentation model is designed for and derived from the machine translation application. We enhance the trivial approach, the segmenting of words in the preprocessing, by integrating the word segmentation into the word alignment training as well as the decoding for the best translations. Translation on Chinese characters is therefore feasible as the segmentation process is pushed to the translation step. In Chapter **??** we will discuss how to combine CWS with training and decoding in Section **??** and Section **??**, respectively.

In the model training we propose a Bayesian semi-supervised Chinese word segmentation model, which uses both monolingual and bilingual information to derive a segmentation suitable for MT. Word segmentation and alignment are trained jointly, and new word entries and their distributions are introduced automatically using linear interpolation. The experiments on both large (GALE) and small (IWSLT) data tracks of Chinese-to-English translation show that the proposed method improves the performance of a state-of-the-art machine translation system.

In the decoding, the translation is performed on the character level. Multiple segmentations are generated as a lattice based on the lexicon obtained from semi-supervised CWS in training. Segmentation decisions are integrated into the search for the best translation. The translation is performed on segmentation alternatives instead of on the single-best segmentation, in order to avoid OOVs and to minimize translation errors. Similar approaches were applied in speech translation, e.g. [?], where speech recognition and text translation are combined by using recognition lattices. We also weight the different segmentations with a language model trained on the Chinese corpus at the word level. Weighting the word segmentation by language model cost was introduced in [?]. The experiments on Chinese-to-English translation show that the proposed method improves the performance of a state-of-the-art machine translation system.

3.2 Sentence segmentation

Another important step is the sentence segmentation. In a statistical machine translation system we define a mathematical model, train the model parameters on the parallel sentence-aligned corpora and translate the test text with this model and its parameters. In practice many sentences in the training corpora are long. Some translation applications cannot handle sentences with a length larger than a predetermined value. The reasons are memory limits and the computational complexity of the algorithms. Therefore, long sentences are usually removed during the preprocessing. But even if long sentences are included, there are still two problems: 1. high computational requirements. 2. the poor quality of the resulting word alignment.

Therefore, we discuss sentence segmentation methods that solve these problems by splitting long sentence pairs into shorter sentence pairs. Previous research on the bilingual sentence segmentation problem can be found in some literature. [**?**] employed 'concept learning' and 'genetic learning' to find potential segmentation positions and to select an actual segmentation point. [**?**] search for the segmentation boundaries using a dynamic programming algorithm. This technique is based on the lexicon information, but only monotone sentence alignments are allowed, and manually defined anchor words are needed as possible segmentation boundaries. [**?**] imposed a compositionality constraint on alignments using IBM model 1. [**?**] presented a method to discover parallel sentences in comparable, non-parallel corpora by training a maximum entropy classifier. Inspired by the phrase extraction approach of [**?**], we introduced a new sentence segmentation method which does not need anchor words and allows for non-monotone alignments of the sub-sentences. As described in [**?**] and here in Section **??**, we separate a sentence pair recursively into two sub-pairs with the IBM model 1 until the lengths of all sub-segments are smaller than a given value. This simple algorithm leads to a significant improvement in translation quality and a speed-up of the training procedure.

Sentence segmentation is the problem of dividing a string of words into sub-strings or a pair of word strings into pairs of sub-strings, while sentence alignment is the problem of making the relations explicit that exist between the sentences of two texts that are known to be mutual translations. The task of sentence segmentation is similar to the task of sentence alignment. In the case of sentence segmentation we assume that the sentence pairs are aligned correctly. The task is to find appropriate split points and to align the sub-sentences. In the case of sentence alignment the corpus is only aligned at the document level. Here, we have to align the sentences of two documents rather than find appropriate split points.

Large-scale translation systems require an automatic and accurate exploitation of bilingual sentences. Therefore, sentence alignment is necessary. The conventional method to align sentences is to break the documents into sentences using anchor words such as punctuation marks and to align these sentences using dynamic programming. Previous investigations can be found in [**?**, **?**, **?**, **?**, **?**] and [**?**]. However, most methods only allow monotone sentence alignment. We will present an extended approach based on [**?**] in Section **??** allowing sentence alignment reordering. This method does not require anchor words and is especially suitable dealing with speech transcriptions. Moreover, it is feasible to combine this approach with other alignment approaches. [**?**] performs a two-stage procedure. The

documents are first chunked and aligned at sentence level using dynamic programming, the initial alignments are then refined to produce shorter segments using binary segmentation. However, on the Chinese-English FBIS training corpus the alignment accuracy and recall are lower than with the Champollion tool [**?**].

We will present a method that leads to more satisfying results. The corpora produced using two approaches are concatenated, and each corpus is assigned a weight. During the training of the word alignment models, the counts of the lexicon entries are linearly interpolated using the corpus weights. In the experiments on the Chinese-English FBIS corpus, the algorithm derived from the method of [**?**] was capable of producing translation results comparable to the Champollion sentence aligner. Using a combination of these two approaches improves the translation performance in comparison to the performance of Champollion.

3.3 Phrase pair segmentation

State-of-the-art statistical machine translation uses phrases as translation units to incorporate context into translation models, as described by [**?**], [**?**] and [**?**]. A phrase is defined as a contiguous sequence of words, a phrase pair contains a phrase in the source language and its translation in the target language. The basic idea of phrase-based translation is to segment the given source sentence into phrases, then translate each phrase and finally compose the target sentence from these phrase translations. Extraction and evaluation of proper phrase pairs are hence crucial for building up a high-performance translator.

The mostly applied phrase pair extraction method is the so-called **Viterbi Extract** [**?**]. We take the Viterbi word alignment of one sentence as input. A source and a target phrase are considered to be translations of each other, if their words are only aligned within this phrase pair and not to the words outside. Summing up phrases and their corresponding translations extracted on all sentences in the bilingual training corpus, we obtain a phrase table with a set of phrase pairs as well as rating scores indicating their significance and translation accuracy.

The translation system employs a phrase-based log-linear model. The decoder generates target sentences from left to right by covering source phrases in a certain order. The underlying feature functions include relative frequencies of the phrase pairs obtained from the extraction process, a word-based lexicon model, a target language model, a heuristic source phrase reordering model, as well as a word and phrase penalty model.

Currently this is the most widely applied phrase pair generation and scoring approach currently. However, it is still far away from a fair and comprehensive evaluation on phrase entries. The frequency model depends heavily on the Viterbi word alignment input, and a mistake in the Viterbi alignment results in errors in the phrase pair extraction. The information from the alignment models is not fully explored and is discarded after the output of the Viterbi word alignment.

Other approaches have been investigated to obtain phrase pairs in less heuristic ways. [**?**] presented an integrated phrase segmentation/alignment algorithm (ISA) for statistical machine translation, which segments and aligns phrases simultaneously. Without the

need of training word alignment models, phrases are identified based on the similarities of mutual information values among word alignment points. [?] presented a technique that begins with improved IBM models to create phrase level knowledge sources that represent effectively local as well as global phrasal context. [?] showed an ITG inspired phrase extraction method using sentence splitting. [?] introduced a joint probability model for statistical machine translation, which automatically learns word and phrase equivalents from bilingual corpora. [?] described a phrase to phrase HMM model to align documents to abstracts. [?] employed posterior probabilities to derive word to phrase as well as phrasal alignments.

In Chapter ?? we will introduce a novel phrase pair induction method by building a mixture model that combines different phrase pair probabilities derived directly from the underlying word alignment models instead of from the Viterbi word alignments. Phrase pair probabilities calculated such as by [?] and [?] are taken as features in the log-linear model. This relaxes the standard phrase extraction heuristics and provides a more flexible infrastructure to introduce phrase pairs from various resources. Learning is pushed down to the level of phrase extraction. All knowledge sources, such as probabilities derived by IBM model 1, HMM and other models, are treated as feature functions in the mixture model framework, which can be extended easily by adding new feature functions. Hence, the standard phrase extraction is a special case by setting the weights of the other features to zero. Additional phrases that are not generated from the standard approach can be obtained, and less meaningful phrase pairs are pruned. The experimental results on the IWSLT 2007 task show that the proposed method improves the translation performance over the conventional method.

3.4 Document segmentation

In SMT models are trained from parallel and monolingual corpora. The quality and quantity of the data and the underlying modeling approach determines the quality of the translation output. With the increasing availability of parallel corpora and better modeling approaches, a significant improvement of the translation quality has been achieved in recent years.

While translation performance has advanced substantially in general, translation style and domain issue leave much room for further improvements. For instance, translating an utterance can be quite different from translating a written sentence in selecting words and phrases and their order. Short phrases such as 'what's up' are more likely to be observed in an informal situation than in written form. This offers a challenge to genre adaptation of SMT systems but at the same time causes a rise to potential improvement if the issue is handled properly. There are mainly four problems to be studied for domain issues.

1. Obtaining in-domain data

   Available parallel or even monolingual domain specific data are usually of limited size. However, building a domain dependent systems requires plenty of training data

with a content closer to the test corpus. Therefore, obtaining large-scale in-domain data from an out-of-domain corpus is a first step for domain adaptation.

Many investigations have been performed for this purpose recently. [?] selected sentences similar to the test set to form an adapted training corpus, which allows a better use of additionally available out-of-domain training data or finds in-domain data in a mixed corpus. [?] use an in-domain translation dictionary and/or in-domain monolingual corpus to improve the in-domain performance. [?] take advantage of a resource-rich language such as English, utilizing cross-lingual information retrieval to adapt language models for a resource-deficient language. Small but effective texts are selected by information retrieval for adaptation. [?] explored transductive learning, where they translated source sentences from the development set and test set repeatedly. The generated translations are then used to improve the performance of the SMT system. [?] proposed methods for feature weight combination and mixture language model adaptation to distinguish multiple domains, which share a general translation engine with phrase-based log-linear models. Domain adaptation during decoding is approached with source text classification methods. A similar approach is presented in [?], where two basic settings are investigated: cross-domain adaptation, in which a small sample of parallel in-domain text is assumed; and dynamic adaptation, in which only the current input source text is considered. Adaptation relies on mixture models estimated on the training data by unsupervised clustering methods. Given available adaptation data, mixture weights are re-estimated adhoc. This work is related to [?]. In addition, we performed unsupervised document clustering to automatically segment a training corpus into different domains.

2. Model combination (translation model, language model)

   Domain specific translation or language models are usually combined so that different models contribute more or less to the final model depending on how its content is related to the test domain. Language model adaptation has been investigated previously in many ways. For example, linear interpolation of a general and a domain specific model by [?]. Back off of domain specific probabilities with those of a specific model [?]. Retrieval of documents pertinent to the new domain and training a language model on-line with this data by [?] and [?]. Maximum entropy, minimum discrimination adaptation by [?]. Adaptation by linear transformation of vectors of bigram counts in a reduced space by [?]. Smoothing and adaptation in a dual space via latent semantic analysis modeling long-term semantic dependencies and trigger combinations by [?]. For the application of statistical machine translation language model interpolation is widely applied, such as in [?] and [?]. While [?] presented an empirical study of four techniques for adapting language models, including a maximum a posteriori (MAP) method and three discriminative training models in the application of Japanese Kana-Kanji conversion. [?] introduced a statistical formulation in terms of a simple mixture model and presented an instantiation of this framework to maximum entropy classifiers and their linear chain counterparts. They presented efficient inference algorithms for this special case based on the technique of conditional expectation maximization. However, further research on these methods in machine translation has not been successful yet. Therefore, we will employ the interpolation techniques for model combination and

direct the attention more to data clustering and model weights optimization in this thesis.

3. Parameter optimization and evaluation set adaptation

   [?] presented the algorithm of boosting and perception as well as the minimum sample risk method for parameters optimization. However, the experiment was not performed for machine translation application. We investigate two optimization methods here: downhill simplex method with the optimization criterion of the final translation performance and the EM algorithm with respect to language model perplexity.

   After building up the domain specific translation systems, we need to know to which domain the evaluation set or part of the evaluation set belongs. Measuring the distance between the evaluation corpus and the domain specific training corpus is therefore a way to classify a test corpus into a domain so that the translation can be performed using the right domain dependent translation system. We will show methods presented in [?] based on information retrieval and language model perplexity. Similar approaches can also be found in [?], where the correlation between the classification error rate and translation performance was not investigated.

In Chapter ?? we will describe domain adaptation in machine translation with unsupervised clustering methods in detail. Solutions to the above mentioned problems are presented including: how to classify training data into distinct domains; how to build domain specific SMT systems in training, and how to optimize model weights and to perform domain adaptation during decoding. For the first problem we propose hierarchical clustering to segment training corpora into multiple parts; for the second problem we use the domain dependent language modeling mixing language models trained on various data; for the last problem, when translating a test document, we will identify its domain automatically and then apply a corresponding decoding setup. Furthermore, different text classification methods will be compared, and their impact on translation performance are investigated. We will see that the domain adaptation using language models does not only enhance the translation performance but also reduces the computational requirements.

Chapter 4

Chinese word segmentation

4.1 Problem description

In Chinese texts, sentences are written as a sequence of Chinese characters without separating the words composed of single or multiple characters with delimiters such as white space. This is different from most European languages and poses a challenge in natural language processing tasks such as machine translation. The conventional way of solving this problem is to segment the Chinese character sequence into Chinese 'words'. Finding proper word boundaries in a sequence of Chinese characters is referred to as the *Chinese word segmentation* problem.

It is difficult to define a 'correct' Chinese word segmentation (CWS), and various definitions have been proposed. In this work, we explore the idea that the best segmentation depends on the task and concentrate on developing a CWS method for MT. For MT, intuitively the best Chinese words should be those units that provide the best word alignment and lead to the best translation performance.

4.2 Definition of Chinese word segmentation

In statistical machine translation we are given a Chinese sentence as a sequence of characters $c_1^K = c_1 c_2 ... c_k ... c_K$ $(k \in 1, 2, .., K)$, which is to be translated into an English sentence of words $e_1^I = e_1 e_2 ... e_i ... e_I$ $(i \in 1, 2, .., I)$, where K and I is the length of the Chinese sentence in characters and the English sentence in words, respectively.

In order to obtain a more adequate mapping between Chinese and English translation units, c_1^K is usually segmented into words. The positions of *Chinese word boundaries* on a character sequence c_1^K is indicated by $k_0 \equiv 0$ and $k_1^J = k_1 k_2 ... k_j ... k_J$ $(j \in 1, 2, .., J)$, where $k_j \in \{1, 2, ...K\}$, $k_{j-1} < k_j$, $k_J \equiv K$, and J is the number of Chinese words in the sentence. We use k_j to indicate that the j-th word segmentation boundary is placed after (on the right side of) the Chinese character c_{k_j} and in front of character c_{k_j+1}, where $1 \leq j < J$. k_0 is a boundary on the left side of the first Chinese character c_1, which is defined as 0, and k_J is always placed after the last Chinese character c_K and therefore

equals K.

Given a sequence of Chinese characters c_1^K and its segmentation boundaries k_1^J, a sentence can also be represented in the form of a sequence of Chinese words $f_1^J = f_1 f_2 ... f_j ... f_J$ $(j \in 1, 2, .., J)$, and each individual Chinese word f_j is defined as

$$f_j = c_{k_{j-1}+1} .. c_{k_j} = c_{k_{j-1}+1}^{k_j} \tag{4.1}$$

f_1^J is composed of two sources of information, the character sequence and its word segmentation.

Table 4.1: An example for the definition of Chinese words and word segmentations.

Characters	小	孩	玩	纸	牌
Glosses of characters	small	child/children	play	paper	card/cards
c_1^K	c_1	c_2	c_3	c_4	c_5
Segmentation boundaries	小	孩	玩	纸	牌
k_1^J		$k_1 = 2$	$k_2 = 3$		$k_3 = 5$
Characters into words		$c_1 c_2$	c_3		$c_4 c_5$
Words		小孩	玩		纸牌
Glosses of words		child/children	play		card/cards
f_1^J		f_1	f_2		f_3

An example is illustrated in Table **??**. The sentence 小孩玩纸牌 contains five Chinese characters ($K = 5$), where c_1 denotes 小, c_2 denotes 孩, etc. The first word segmentation boundary is placed after the second Chinese character ($k_1 = 2$), the second boundary is after the third character ($k_2 = 3$) and the third boundary is after the fifth character ($k_3 = 5$). That means 小(c_1) and 孩(c_2) together compose the word 小孩(f_1), 玩(f_2) is a single character word, and 纸牌($c_4 c_5$) is the third word (f_3) of this sentence.

4.3 Common methods

We will give a short overview on the current Chinese word segmentation methods in statistical machine translation. These methods can be categorized into three types:

1. Each Chinese character is treated as a word.

   The Chinese training and test corpora are processed with their single characters, and each single character is considered as a word. That means for a character sequence c_1^K, $J = K$ and $k_j = j$ and $f_j = c_j$ for all $j \in \{1, ..., J\}$. Training and translation at the Chinese character level do not require additional tools or human effort. But [**?**] showed that direct translation on the character level does not lead to the same translation performance as translation on the word level.

2. The training and test texts are segmented manually.

   The word segmentation boundaries k_1^J are decided by humans. Manual segmentation avoids the segmentation errors but requires human effort. Moreover, the correct segmentation does not always lead to the best translation result. The segmentation in the test set should be consistent with the segmentation in the training to avoid out-of-vocabulary words.

3. The training and test texts are segmented by an automatic segmentation tool.

   This is a widely applied solution to the Chinese word segmentation problem. The Chinese texts are segmented using an off-the-shelf Chinese word segmentation method and translated afterwards given this fixed segmentation. This approach usually performs better than the previous two methods.

4.4 n-gram segmentation

The simplest and widely applied automatic segmentation tool is based on a *unigram segmentation*, which requires a manual lexicon containing a list of Chinese words and their frequencies, an example of which is shown in Table **??**. The lexicon and frequencies can be obtained using manually annotated data, e.g. the LDC lexicon [**?**] or a lexicon that is extracted from the alignment of the training corpora [**?**]. We need to maximize the probability of a sentence considering all word segmentation alternatives. Assuming each Chinese word in the sentence is distributed independently, we are interested in knowing how to put the word delimiters properly so that the product of the probabilities of all words is maximized:

$$\hat{k}_1^{\hat{J}}(c_1^K) = \underset{k_1^J, J}{\operatorname{argmax}} \prod_{j=1}^{J} Pr(f_j = c_{k_{j-1}+1}^{k_j}) \tag{4.2}$$

$$= \underset{k_1^J, J}{\operatorname{argmax}} \prod_{j=1}^{J} Pr(c_{k_{j-1}+1}^{k_j}) \tag{4.3}$$

$Pr(f_j = c_{k_{j-1}+1}^{k_j})$ is the probability of a word f_j which is a sequence of characters $c_{k_{j-1}+1}^{k_j}$ with a boundary after the k_{j-1}-th character and before the k_j-th character in the sentence. Taking the word dependency into account we extend Equation **??** into Equation **??**, by using the concept of an n-gram language model, where n is the order of the n-gram model:

$$\hat{k}_1^{\hat{J}}(c_1^K) = \underset{k_1^J, J}{\operatorname{argmax}} \prod_{j=1}^{J} Pr(f_j{=}c_{k_{j-1}+1}^{k_j} | f_{j-1-n}{=}c_{k_{j-2-n}+1}^{k_{j-1-n}}, .., f_{j-1}{=}c_{k_{j-2}+1}^{k_{j-1}}) \tag{4.4}$$

$$= \underset{k_1^J, J}{\operatorname{argmax}} \prod_{j=1}^{J} Pr(c_{k_{j-1}+1}^{k_j} | c_{k_{j-2-n}+1}^{k_{j-1-n}}, .., c_{k_{j-2}+1}^{k_{j-1}}) \tag{4.5}$$

Table 4.2: All possible word segmentations for c_1^K are illustrated in Table **??**. The number of all possible segmentations for c_1^K is 2^{K-1}. The best word segmentation is the sentence with ID. 2.

ID.	J	Chinese word sequence
1	5	小 孩 玩 纸 牌
2	3	小孩 玩 纸牌
3	4	小孩 玩 纸 牌
4	3	小孩 玩纸 牌
5	2	小孩 玩纸牌
6	3	小孩玩 纸 牌
7	2	小孩玩 纸牌
8	2	小孩玩纸 牌
9	1	小孩玩纸牌
10	3	小 孩玩 纸 牌
11	2	小 孩玩 纸牌
12	3	小 孩玩纸 牌
13	2	小 孩玩纸牌
14	4	小 孩 玩纸 牌
15	3	小 孩 玩纸牌
16	4	小 孩 玩 纸牌

Table 4.3: Manually generated Chinese word lexicon.

Word	小	孩	玩	纸	牌	小孩	...
Frequency	3465	22	588	146	361	367	...

The dynamic programming algorithm is used to maximize the product of the relative frequencies of all words in one sentence. The segmenter finds the path which has the highest product of word probabilities.

Another instance of this type of segmenter is the LDC tool, which is also based on unigram segmentation but with additional text normalizations. The LDC segmenter finds the path which has the highest product of word probability and the next word is selected from the longest phrase. More details can be found on the LDC web site [**?**].

The unigram Chinese word segmentation method is so far the most commonly applied method in machine translation, but it has several problems: First, maximizing the product of single word probabilities does not guarantee that the context information is taken into account; hence, the segmentation may contain errors. Second, a more accurate word segmentation does not always lead to a great improvement in translation performance. The "correct" segmentation for one character sequence is not universal but depends on the Chinese context and the destination language.

This method is sub-optimal for MT. For example, 纸(paper) and 牌(card) can be separated or composed into one word 纸牌(cards). Because 纸牌 does not exist in the manual lexicon, it cannot be generated by this method.

Table 4.4: An example of translation hypotheses of a Chinese sentence with different Chinese word segmentations.

In Characters	小 孩 玩 纸 牌
Segmentation 1.	小孩 玩 纸 牌
Segmentation 2.	小孩 玩 纸牌
...	...
Reference	Children play cards
Hypothesis 1.	Children play a card
Hypothesis 2.	Children play cards

An example is shown in Table **??**: the first line is a Chinese sentence as a sequence of characters selected from the NIST 2006 translation evaluation set. Using the LDC first segmentation method described in Section **??**, "'纸 牌"' is separated into two words, and the translation 'a card' is incorrect. In contrast, the second segmentation method that we will describe in this chapter leads to a correct translation.

In standard approaches word segmentation is performed beforehand and independent of the translation system. Segmentation and alignment of words are two separate processes, though they actually influence each other. In the widely applied unigram method, single word probabilities are computed as relative frequencies using a manually generated lexicon. But this lexicon requires human effort and might be sub-optimal for certain tasks. As we know, the definition of correct Chinese word segmentation depends on the application. The goal of this work is to find an optimal CWS for Chinese-English MT. We assume that for MT the best Chinese words are those translation units that provide us with the best word alignment and phrase table, which leads to best translation performance. We evaluate the translation performance in this chapter using BLEU [**?**] and TER [**?**]. The proposed approach outperforms other methods in two aspects: First, employing the Bayesian approach allows us to introduce new words to the lexicon with a prior distribution. Second, the semi-supervised training algorithm jointly optimizes CWS and word alignment discriminatively with respect to translation performance.

We developed novel methods that learn the word distributions and even word entries in the lexicon automatically from the bilingual training corpus. Chinese word segmentation and word alignment are trained jointly. In Section **??** we will introduce the learned segmentation method to train a word segmentation model using the word alignment information. The words in the texts are hence segmented without any pre-defined human knowledge. An unsupervised CWS method will be described in detail, where both word entries and word distributions are learned simultaneously. The model parameters are optimized with respect to the translation performance.

We will describe advanced methods for CWS in detail. In the model training we apply two approaches: learned segmentation and semi-supervised word segmentation. In learned segmentation a lexicon such as in Table **??** is extracted automatically from single-best word alignments trained on Chinese characters and English words. Bilingual and context information are employed with respect to the translations. However, the single-best word alignments can contain errors, thus we further refined the model into a semi-supervised word segmentation method, where word segmentation and alignment are trained jointly using both monolingual and bilingual information; see [**?**]. In the translation process multiple segmentations are represented as a lattice so that segmentation decisions are integrated into the search for the best translation, as presented in [**?**]. Multiple segmentations are incorporated into the translation instead of a single-best segmentation.

4.5 Segmentation learned from the alignment

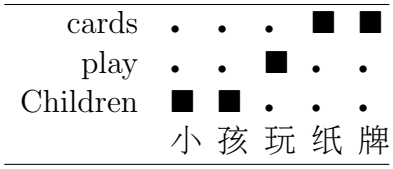


Figure 4.1: An example of an alignment matrix between Chinese characters and English words. A black box indicates a single-best (Viterbi) word alignment.

We will introduce the first word segmentation approach in this section, namely learned segmentation. In statistical machine translation, a bilingual corpus is used. As introduced by [**?**], from this corpus a segmentation of the Chinese part is obtained in the following way: First, we train the statistical translation models on the bilingual corpus. There is no word segmentation performed on Chinese texts, and each Chinese character is interpreted a word, as described for the first segmentation type in Section **??**.

As a result of this word alignment training, we obtain for each sentence pair a mapping of Chinese characters to the corresponding English words i.e. the single-best word alignment between Chinese characters and English words. Such an alignment is represented as a binary matrix with $K \cdot I$ elements. An example is shown in Figure **??**, where a Chinese training sentence in characters is plotted along the horizontal axis and its English translation sentence in words along the vertical axis. The black boxes show the best alignment for this sentence pair after word alignment training. In this example the first two Chinese tokens are aligned to "Children", the next one is aligned to "play", and the last two tokens are aligned to "cards".

Based on this information, we can generate a Chinese word list with each entry composed of one or more Chinese characters, which are aligned to one English word in the word alignment matrix. In the experiments we train the alignments in both directions with the GIZA++ tool and combine them. Chinese word entries are extracted based on this combined alignment. Lexicon entries learned from the alignment matrix in Figure **??** are '小孩', '玩' and '纸牌'; each of them has a frequency of one. With the help of this self-learned lexicon we use a segmentation tool, such as a unigram segmenter in Section **??** to obtain a segmented Chinese text. If we calculate the frequencies for every word, the distribution can be obtained, too. Finally, we retrain the translation system with the segmented corpus.

This lexicon shows the distribution of Chinese words in the training corpus. The extraction method differs from other self-learned methods because it uses the bilingual training corpus instead of the monolingual corpus such as in [**?**]. Since we are more interested in the relationship between the languages, this method is more suitable for the machine translation application.

The central idea of the proposed lexicon learning method is: several Chinese characters constitute a Chinese word if they are aligned to the same English word. Using this idea and the bilingual corpus, we can generate a Chinese lexicon automatically. As a conclusion, the 'learned translation with learned segmentation' consists of three steps:

1. The input is a sequence of Chinese characters without segmentation. After the training using GIZA++ , we extract a monolingual Chinese dictionary from the alignment.

2. Using this learned dictionary we segment the sequence of Chinese characters into words. In other words, the unigram method is used, but the manual lexicon is replaced by the learned lexicon.

3. Based on this word segmentation, we perform another training using GIZA++ . Then, after training the models IBM model 1, HMM and IBM model 4, we extract bilingual word groups, which are referred to as phrase pairs.

4.6 Semi-supervised Chinese word segmentation

The learned segmentation method models the word distributions in the lexicon using alignment information. But an erroneous alignment can result in an incorrect word segmentation, which may lead to sub-optimal translation results. Therefore, we further propose a refined Chinese word segmentation model that learns both Chinese word entries and their distribution to generate a dynamic lexicon. New words are introduced with a prior distribution using Bayesian learning. The Chinese word segmentation and the word alignment, which influence each other, are trained simultaneously.

This method is semi-supervised, namely, the Chinese word segmentation is jointly trained with the word alignment given an initialized word segmentation and alignment. We employ linear interpolation to introduce new words to the lexicon with a prior distribution. We describe a generative model which consists of a word model and two alignment models, representing the monolingual and bilingual information respectively. We first segment the Chinese text using a unigram segmenter and then learn new words and word distributions, which are suitable for MT.

The experiments show that both in a large (GALE) and in a small (IWSLT) data track of Chinese-to-English translation, the proposed method improves the performance of a state-of-the-art machine translation system.

In the following text, we will first introduce the semi-supervised word segmentation approaches in Section **??**, along with their main characteristics that are different from the other approaches. Then we will describe the generative model in Section **??** in detail, which is the central idea of this approach. Furthermore, in Section **??** we will extend the generative model to a final model, similar to a maximum entropy model in which most features are derived from the sub-models of the generative model. Finally, we will discuss the training procedures using Gibbs sampling algorithm and the re-alignment alternatives in Section **??**, **??** and **??**.

4.6.1 Approaches

The training and translation processes for semi-supervised Chinese word segmentation of the above example sentence are illustrated in Figure **??**. The inputs to the system are the bilingual training data, including Chinese sentences in characters and its English translations in words, a manual Chinese word lexicon, such as the LDC lexicon, as well as a test corpus on the character level.

First, we segment the character sequence of the Chinese training corpus with a unigram word segmentation using the manual lexicon, then the word alignment and Chinese word segmentation are jointly trained using semi-supervised Chinese word segmentation. After that, by counting the Chinese word frequencies of the training corpus, we easily obtain an automatically generated lexicon. A lexicon combining the automatic and the manual lexicons is then applied to perform a unigram segmentation on the test corpus. The Chinese word segmentation on the test corpus is another output from the proposed segmentation system.

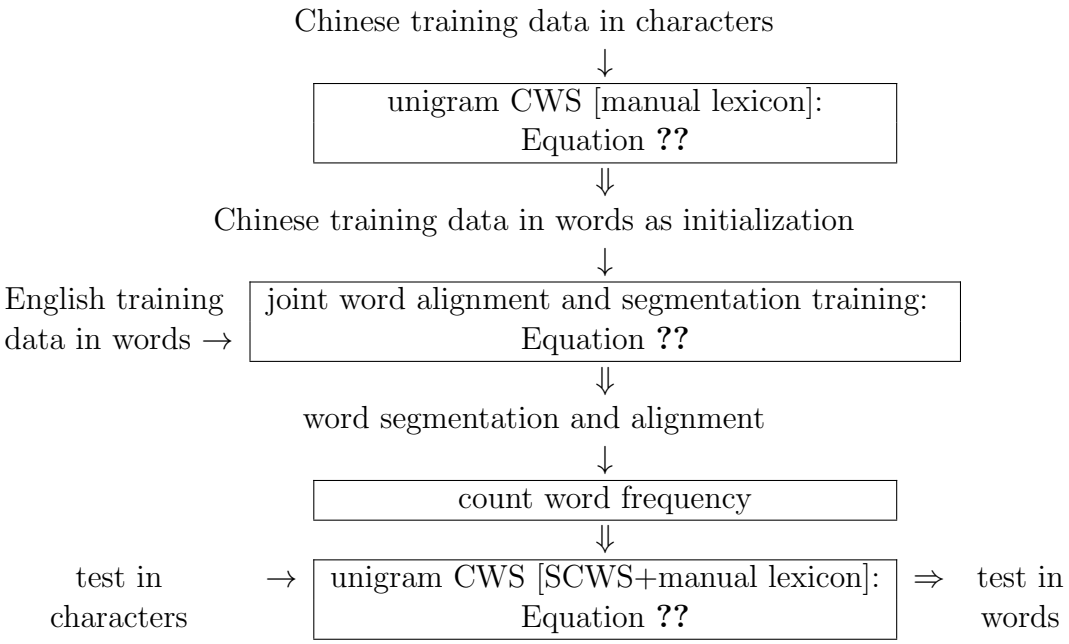


Figure 4.2: Workflow of semi-supervised CWS.

There are two main techniques for Bayesian estimation of such models: Markov Chain Monte Carlo (MCMC) and Variational Bayes (VB). MCMC encompasses a broad range of sampling techniques including component-wise Gibbs sampling, named after the physicist J. W. Gibbs. In general, MCMC techniques do not produce a single model that characterizes the posterior, but instead produce a stream of samples from the posterior.

Gibbs sampling is a widely applicable MCMC, which generates a sequence of samples from the joint probability distribution of two or more random variables. The purpose of such a sequence is to approximate the joint distribution or to compute an integral (such as an expected value). This algorithm is applicable if the joint distribution is not known explicitly, but the conditional distribution of each variable is known. The Gibbs sampling algorithm generates an instance from the distribution of each variable in turn, depending on the current values of the other variables. This characteristic is particular interesting if the categories are unknown and to be learned. In this chapter the categories are the Chinese words, namely word entries themselves. There are approximately 90K Chinese characters and 7000 are commonly in use; any of these characters can be an element of a word. Usually, a Chinese word is composed of one to four characters. The number of Chinese words is calculated as $7000 + 7000^2 + 7000^3 + 7000^4$, which is difficult to be fixed before translation. If there is no previously defined lexicon, the number of all possible segmentations for a sequence of characters c_1^K is 2^{K-1}, as illustrated in Table **??**. That means that the complexity is exponential in the order of character sequence length. Therefore, we have to approximate the space of all possible derivations in some way: we can define that a word contains at most four characters. With such a constraint, the complexity becomes polynomial, but it is still a high order polynomial. We can perform a standard beam search to prune low cost paths. As an alternative to draw a space with all segmentation derivations for one sentence, we apply the Gibbs sampling algorithm, which

learns each parameter value depending on all other parameter values in turn. [**?**] showed that in the decoding the search spaces produced by the sampling approach occupied roughly half the memory as those produced by the beam search with similar results.

4.6.2 Generative model

Table 4.5: Observations and hidden variables of the generative model for Chinese word segmentation.

	Symbol	Abb.	Example
Observations			
Chinese sequence in characters	c_1^K	C	小 孩 玩 纸 牌
English sentence	e_1^I	E	Children play cards
Hidden variables			
Alignment normal	a_1^J	A	e.g. cards → 牌
Alignment inverse	b_1^I	B	e.g. 牌→ cards
Segmentation (Chinese sequence in words)	k_1^J (f_1^J)	(F)	e.g. 小孩 玩 纸牌

We apply a generative classifier to learn a model of the joint probability of observations. As shown in Table **??**, the generative model assumes that a corpus of parallel sentences (${c_1}^K$,${e_1}^I$) is generated along with a hidden sequence of Chinese words ${f_1}^J$ and a hidden word alignment ${b_1}^I$ for every sentence. The alignment indicates the aligned Chinese word f_{b_i} for each English word e_i, where f_0 indicates a special *null* word as in the IBM models.

The joint probability of the observations $({c_1}^K, e_1^I)$ can be obtained by summing up all possible values of the hidden variables k_1^J and b_1^I. The model probability $Pr(c_1^K, e_1^I)$ can be seen as the sum of all possible Chinese word segmentations k_1^J of the character sequence c_1^K:

$$Pr(c_1^K, e_1^I) = \sum_{k_1^J} \sum_{b_1^I} Pr(c_1^K, e_1^I, k_1^J, b_1^I) \tag{4.6}$$

Given a sequence of Chinese characters and its word segmentation boundaries, the corresponding sequence of Chinese words is determined, and vice versa. Given a sequence of Chinese words, we can determine its sequence of Chinese characters and its word segmentation boundaries. Therefore, we use f_1^J to represent the information of c_1^K and k_1^J. Without assuming any special form for the probability of a sentence pair along with hidden variables, we can factor it into a monolingual Chinese sentence probability and a bilingual translation probability as follows:

$$Pr(e_1^I, f_1^J, b_1^I) = Pr(f_1^J) Pr(e_1^I, b_1^I | f_1^J) \tag{4.7}$$

This suggests a monolingual Chinese sentence model $Pr(f_1^J)$ and a bilingual translation model $Pr(e_1^I, b_1^I|f_1^J)$. In the following paragraphs we will describe the modeling assumptions behind the monolingual Chinese sentence model and the translation model, respectively.

4.6.2.1 Monolingual Chinese sentence model

We use the unigram model to estimate the sentence probability using monolingual information. In this model words are generated independently. The probability of a sequence of Chinese words in a sentence is thus:

$$Pr(f_1^J) = \prod_{j=1}^{J} P_G(f_j), \tag{4.8}$$

where $P_G(f_j)$ is further explained by the word generation model in the following section.

4.6.2.2 Word generation by Bayesian learning

The conventional Chinese word segmentation approach applies a manual lexicon containing fixed Chinese words and their frequencies as distributions. Different from a standard method, the semi-supervised segmentation model can introduce new Chinese words and learn word distributions automatically from unlabeled data.

We never estimate a word distribution explicitly but instead integrate over its possible values and perform a Bayesian inference. It is easy to compute the probability of a Chinese word given a set of already generated words. According to this model, each word in a Chinese corpus is generated using linear interpolation:

$$P_G(f) = (1 - \alpha_1)\frac{N(f)}{N} + \alpha_1 P_0(f) \tag{4.9}$$

This is done by casting Chinese word generation as a Chinese restaurant process [**?**] i.e. a restaurant with an infinite number of tables (approximately corresponding to Chinese word types), each table with infinite number of seats (approximately corresponding to Chinese word frequencies). The model with linear interpolation in Equation **??** is equivalent to a word model based on the Chinese restaurant process, such as the word model in [**?**]. Each random variable f is drawn independently and identically distributed from G , where G is a distribution over words drawn from a prior with base measure P_0 and a concentration parameter.

$N(f)$ is the number of Chinese words f in the previous context: In the first training iteration, $N(f)$ is the frequency of word f appearing from the beginning of the text to the current position; after the first iteration, it is the frequency of word f in the text counted in the last iteration. the N is the total number of Chinese words, P_0 is the base probability over words, and α_1 influences the probability of introducing a new word at

each step and controls the size of the lexicon. The probability of generating a word from the cache increases as more instances of that word are seen. α controls the number of word types, i.e. size of the lexicon. It is the total probability to generate any new word. P_0 defines a probability distribution over new words i.e. how likely a sequence of Chinese characters forms a word.

For the base distribution P_0, which governs the generation of new words, we use the following distribution (called the **spelling model**):

$$P_0(f) = \sum_L P(L)P(f|L), \tag{4.10}$$

where L is the number of Chinese characters of word f. We decompose the spelling model into a word length model $P(L)$ and a word model depending on its length $P(f|L)$. The length model follows a Poisson distribution:

$$P(L) = \frac{\kappa^L}{L!}e^{-\kappa} \tag{4.11}$$

According to Poisson, $\sum_L P(L) = \sum_L \frac{\kappa^L}{L!}e^{-\kappa} = 1$.

The word model is defined based on two cases: a uniform distribution on words with a length equals to the given length and zero otherwise. The probability of a word f given a word length L is defined as

$$P(f|L) = \begin{cases} (\frac{1}{|c|})^L & : |f| = L \\ 0 & : |f| \neq L \end{cases} \tag{4.12}$$

$|f|$ is the length of the word f, and $|c|$ is the character vocabulary size i.e. the number of different characters in the document. The normalization constraint is proven for Equation **??** as follows: $\sum_f P(f|L) = |c|^L \cdot (\frac{1}{|c|})^L + 0 = 1$, because there are $|c|^L$ possible words with a length of L and each has a probability of $(\frac{1}{|c|})^L$, the other words with a length not equals to L have a probability of 0.

4.6.2.3 Translation model

We employ the inverse IBM model 1 to generate English words and alignments given the Chinese words. In this model, for every Chinese word f (including the *null* word), a distribution over English words G_f is first drawn from a prior $P_0(e)$, which is estimated by the empirical distribution over English words in the parallel data. Then, given these parameters, the probability of an English sentence and alignment given a Chinese sentence (sequence of words) is given by

$$P(e_1^I, b_1^I|f_1^J) = \prod_{i=1}^{I} \frac{1}{J+1} P_{G_{f_{b_i}}}(e_i|f_{b_i}), \tag{4.13}$$

where e_i is distributed according to $G_{f_{b_i}}$. This is the same model form as the inverse IBM model 1. We place priors on the Chinese-word specific distributions over English words. [a]

In practice, we observed that using a word-alignment model in one direction is not sufficient. We then added a factor to the final model which includes word alignment in the other direction. Such combinations of models in both directions are widely used for phrase extraction [**?**].

Therefore, we also used a translation model in the other direction, the IBM model 1. We ignore the detailed description here, because the calculation is the same as that of the inverse IBM model 1. According to this model, for every English word e (including the *null* word), a distribution over Chinese words G_e is first drawn from a prior $P_G(f)$ derived from Equation **??**. The probability of a sequence of Chinese words f_1^J and a word alignment a_1^J given a sequence of English words e_1^I is then:

$$P(f_1^J, a_1^J|e_1^I) = \prod_{j=1}^{J} \frac{1}{I+1} P_{G_{e_{a_j}}}(f_j|e_{a_j}) \tag{4.14}$$

4.6.3 Final model

We put the monolingual model and the translation models in both directions together into a single model, where each of the component models is weighted by a scaling factor. This is similar to a maximum entropy model. We optimize the weights of the sub-models on a development set by maximizing the BLEU score of the final translation. We used three features derived from Equation **??** and equations in Section **??**.

The maximum entropy model can be viewed as a weighted linear combination of the log probabilities of the sub-models. The weights that are optimized on development datasets have empirical justifications. Since different sub-models have been trained on different datasets, their dynamic value ranges can be so different that it is inappropriate to combine their log probabilities through simple addition. Moreover, for instance, some models may be poorly estimated due to the lack of a large amount of training data. Therefore, empirical results have demonstrated that the use of scaling factors that reflect the relative contribution of different sub-models often improves the performance. Similar approaches have been used very successfully before, for example in the IBM models 3–6 [**?**]. The final model applied in the experiments is

$$\begin{aligned}
&(\hat{k}_1^{\hat{J}}(c_1^K, e_1^I), \hat{a}_1^{\hat{J}}(c_1^K, e_1^I), \hat{b}_1^I(c_1^K, e_1^I)) \\
&= \underset{k_1^J, J, a_1^J, b_1^I}{\operatorname{argmax}} \left\{ Pr(c_1^K, k_1^J)^{\lambda_1} Pr(e_1^I, b_1^I|c_1^K, k_1^J)^{\lambda_2} Pr(c_1^K, k_1^J, a_1^J|e_1^I)^{\lambda_3} \right\}
\end{aligned} \tag{4.15}$$

where a is the alignment for the Chinese-to-English translation, and b is the alignment for the English-to-Chinese translation.

[a] f_{b_i} is the Chinese word aligned to e_i and $G_{f_{b_i}}$ is the distribution over English words conditioned on the word f_{b_i}. Similarly, e_{a_j} is the English word aligned to f_j in the other direction and $G_{e_{a_j}}$ is the distribution over Chinese words conditioned on e_{a_j}.

4.6.4 Gibbs sampling training

Using the generative model we would like to choose the most likely word segmentation given the observed pairs of Chinese-English sentences.

It is generally impossible to find the most likely segmentation according to the proposed model using exact inference, because the hidden variables do not allow exact computation of the integrals. Nonetheless, it is possible to define algorithms using Markov chain Monte Carlo (MCMC) that produces a stream of samples from the posterior distribution of the hidden variables given the observations. We applied the Gibbs sampler [**?**], one of the simplest MCMC methods, in which transitions between states of the Markov chain result from sampling each component of the state conditioned on the current value of all other variables.

In this work, the observations are $D = (d_1, ..d_s, .., d_S)$, where d_s=(c_1^K, e_1^I) indicates a bilingual sentence pair, the hidden variables are the word segmentations f_1^J and the alignments in two directions a_1^J and b_1^I.

Gibbs sampling is an iterative procedure that samples variables given the current values of all other variables. The Gibbs sampler for Chinese word segmentation works as follows: For each step, we take a single possible boundary point by fixing other segmentations and alignments, then we compare hypotheses considering this boundary and the related alignments. After sampling by using the posterior probabilities of each candidate, we choose one of these candidates and perform the same operation for the next position.

To perform Gibbs sampling we start with an initial word segmentation and with initial word alignments. We re-sample iteratively the word segmentation and alignments according to Equation **??**.

For example, we are interested in determining the word boundary after 纸 in the sentence '小孩玩纸牌'. We only show the example with the monolingual model for convenience. We suppose that 纸 牌 are two words from the initialization. N is the number of words in Chinese corpus. First, we decrease the related counts N, N(纸), N(牌), N(纸, Children), .. by one. After that, we calculate the probabilities P(纸 牌|..), P(纸牌|..), .. again. Now, we compare P(纸 牌|..) and P(纸牌|..) using sampling i.e. after the normalization on the probabilities so that P'(纸 牌|..) and P'(纸牌|..) sum to one, we select a random number between zero and one, if this random number is smaller than P'(纸牌|..), we choose 纸牌, otherwise, we choose 纸 牌. That means that a higher probability segmentation is more likely to be chosen. Finally, we increase the associate counts of the chosen segmentation N, N(纸牌), N(纸牌, Children), N(纸牌, play), N(纸牌, card),.. by one. This is an iterative process going over all positions in a document until the segmentation results converge.

We only allow limited modifications to the initial word alignments for reasons of efficiency. Thus, we only use models derived from IBM model 1 (instead of IBM model 4) for comparing different word segmentations and not for large-scale modification of the word alignment. IBM model 4 from GIZA++ is an improved model in comparison to IBM model 1 that we use. On the other hand re-sampling the segmentation causes relinking alignment points to parts or groups of the original words.

Hence, we organize the sampling process around possible word boundaries. For each character c_k in each sentence, we consider two alternative segmentations: ${c_k}^+$ indicates the segmentation where a boundary exists after c_k, and ${c_k}^-$ indicates the segmentation where no boundary exists after c_k keeping all other boundaries fixed. Let f denote the single word spanning character c_k if there is no boundary after it, and f',f'' denote the two adjacent words resulting if there is a boundary: f' includes c_k and f'' starts just to the right, with character c_{k+1}. The introduction of f' and f'' leads to $2^{|e|}$ new possible alignments in the E-to-C direction $b^+_{k1}, \ldots, b^+_{k|c|}$, such as in Figure **??**. $|e|$ is defined as the total number of English words aligned to f' or f'', previously to f. Together with the boundary versus no-boundary state at each character position, we re-sample a set of alignment links between English words and any of the Chinese words f or f' and f'' keeping all other word alignments in the sentence pair fixed.

Thus, we consider a set of alternatives for the boundary after c_k and relevant alignment links at each step in the Gibbs sampler keeping all other hidden variables fixed. We need to compute the probability of each of the alternatives at each step given the fixed values of the other hidden variables.

We introduce some notation to make the presentation easier. For every position k in sentence pair s, we denote by dh_{sk} the observations and hidden variables for all sentences other than sentence s, and the observations and hidden variables inside sentence s, not involving character position c_k. The fixed variables inside the sentence are the words not neighboring position k and the alignments in both directions to these words. ${dh_{sk}}^+$ denotes that there is a word boundary after c_k in sentence s, and ${dh_{sk}}^-$ denotes that there is no word boundary after c_k in sentence s given the observations.

In the process of sampling we consider a set of alternatives: segmentation ${c_k}^+$ along with the product space of relevant alignments in both directions $b^+_{k1}, \ldots, b^+_{k|e|}$, and a^+_k, and segmentation c^-_k along with relevant alignments ${b_k}^-$ and a^-_k. For brevity reasons, we denote these alternatives by ${cba_{k,e\_}}^+$ and ${cba_k}^-$, where $e\_$ is the $e\_$-th candidate after re-alignment. Table **??** shows schematically one iteration of Gibbs sampling through the whole training corpus of parallel sentences, where S is the number of parallel sentences.

We will describe how we compute probabilities of alternatives in Section **??** and how we determine the set of alternative hypotheses in Section **??**.

4.6.5 Computing probabilities of alternatives

For the Gibbs sampling algorithm in Table **??**, we need to compute the probability of each alternative segmentation/alignments given the fixed values of the rest of the data dh_{sk}. The probability of the hidden variables in the alternatives is proportional to the joint probability of the hidden variables and observations, and thus it is sufficient to compute only the probability of the latter.

Let cba_k denote an alternative hypothesis including a boundary or no boundary at position k, and relevant alignments to English words in both directions of the one or two Chinese words resulting from the segmentation decision at k. The probability of this configuration given by the proposed model is

Table 4.6: General algorithm of Gibbs sampling for CWS.

1	Input: initial segmentation and alignments
2	Output: sampled segmentation and alignments
3	for each sentence $s = 1$ to S
4	for each character position $k = 1$ to K where $c_k \in d_s$
5	Create $\|e\| + 1$ candidates, $cba^+_{k,e\_}$ and cba^-_k, where
6	$cba^+_{k,e\_}$: there is a word boundary after c_k
7	cba^-_k: there is no word boundary after c_k
8	Compute probabilities
9	$P(cba^+_{k,e\_}\|{dh_{sk}}^-)$
10	$P(cba^-_k\|{dh_{sk}}^+)$
11	Sample boundary and relevant alignments
12	Update counts

$$P(cba_k|dh_{sk}) \propto P_m(cba_k|dh_{sk})^{\lambda_1} \cdot P_{ef}(cba_k|dh_{sk})^{\lambda_2} \cdot P_{fe}(cba_k|dh_{sk})^{\lambda_3}, \tag{4.16}$$

where $P_m(cba_k|dh_{sk})$ is the monolingual word probability, and $P_{fe}(cba_k|dh_{sk})$ and $P_{ef}(cba_k|dh_{sk})$ are the translation probabilities in the two directions.

Now we describe the computation of each of the component probabilities.

4.6.5.1 Word model probability

The word model probability $P_m(cba_k|dh_{sk})$ in Equation **??** is derived from Equation **??**. There are two cases: If the hypothesis specifies that there is a boundary after character c_k, we need the probabilities of the two resulting words f', and f''; otherwise, we need the probability of the single word f.

Let N denote the total number of word tokens in the rest of the corpus dh_{sk}, and $N(f)$ denote the number of instances of word f in dh_{sk}. The probabilities in the two cases $P_m(cba^+_k|dh_{sk})$ and $P_m(cba^-_k|dh_{sk})$ are computed using $P_G(f') \cdot P_G(f'')$ and $P_G(f)$ respectively.

4.6.5.2 Translation model probability

The translation model probabilities depend on whether or not there is a segmentation boundary at c_k. They also depend on the English words which are aligned to the relevant Chinese words.

In the first case, we assume that there is a word boundary in cba_k, but previously there was no word boundary in cba_k (see Figure **??** and **??**). Here, we overload the notation and use b_k and a_k to indicate the alignments of the relevant Chinese words at position k

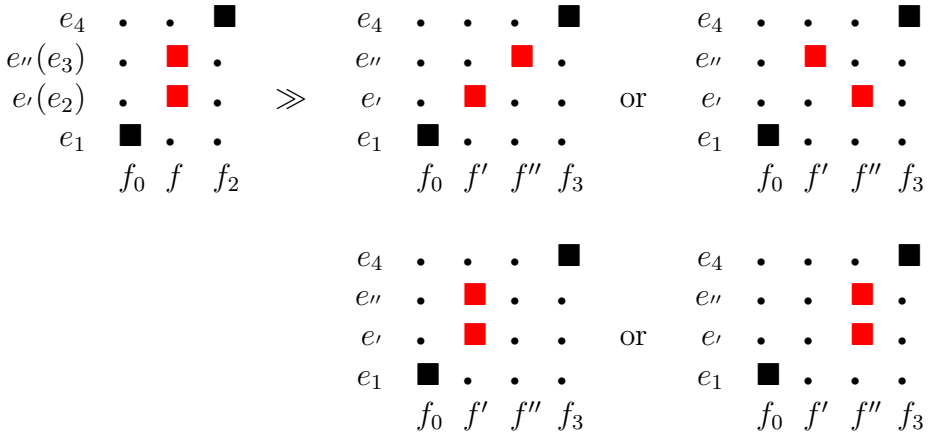


Figure 4.3: Transition from no boundary($-$) to a boundary($+$): The monolingual probability $P_m(cba_k^+|dh_{sk}{}^-)$ is estimated as $P_G(f')P_G(f'')$, and the translation probability in the E-to-C direction $P_{ef}(cba_k^+|dh_{sk}{}^-)$ is estimated as $\frac{1}{J+2}^{|e|}P(e_{'}|f')P(e_{''}|f'')$ or $\frac{1}{J+2}^{|e|}P(e_{''}|f')P(e_{'}|f'')$ or $\frac{1}{J+2}^{|e|}P(e_{'}|f')P(e_{'}|f'')$ or $\frac{1}{J+2}^{|e|}P(e_{''}|f')P(e_{''}|f'')$, here $|e| = 2$ and $J = 2$.

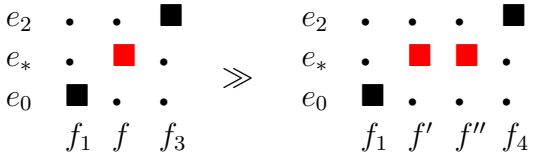


Figure 4.4: Transition from a no-boundary($-$) to a boundary($+$): The monolingual probability $P_m(cba_k^+|dh_{sk}{}^-)$ is estimated as $P_G(f')P_G(f'')$, and the translation probability in the C-to-E direction $P_{fe}(cba_k^+|dh_{sk}{}^-)$ is estimated as $\frac{1}{I+1}^2 P_G(f'|e_*)P_G(f''|e_*)$, here $|e| = 2$ and $I = 2$.

to any English word. Let I denote the total number of English words in the sentence, and J denote the number of Chinese words according to this segmentation. We consider the *null* words when calculating IBM model 1 in both directions. We also denote the total number of English words aligned to either f' or f'' in the E-to-C direction by $|e|$.

The translation model probability from no word boundary to a word boundary in the E-to-C direction if f' aligns to e' and f'' aligns to e'' as in Figure **??** is thus

$$P_{ef}(cba_k^+|dh_{sk}{}^-) : \left(\frac{1}{J+2}\right)^{|e|} P_G(e'|f')P_G(e''|f''). \tag{4.17}$$

The translation model probabilities with other re-alignments in Figure **??** are calculated in the same way. Here we compute $P_G(e'|f')$ and $P_G(e''|f'')$ as

$$P_G(e|f) \quad = \quad (1-\alpha_3)\frac{N(e,f)}{N(f)} + \alpha_3 P(e), \tag{4.18}$$

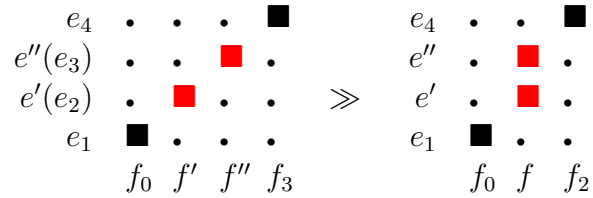


Figure 4.5: Transition from a boundary(+) to no boundary(−): The monolingual probability $P_m(cba_k^-|dh_{sk}{}^+)$ is estimated as $P_G(f)$, and the translation probability in the E-to-C direction $P_{ef}(cba_k^-|dh_{sk}{}^+)$ is estimated as $\frac{1}{J}{}^{|e|}P_G(e'|f)P_G(e''|f)$, here $|e| = 2$,$J = 3$.

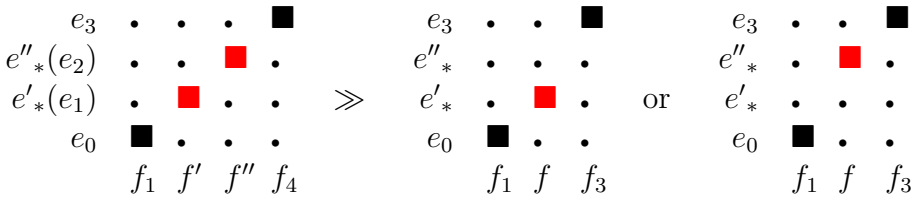


Figure 4.6: Transition from a boundary(+) to no boundary(−): The monolingual probability $P_m(cba_k^-|dh_{sk}{}^+)$ is estimated as $P_G(f)$, and the translation probability in the C-to-E direction $P_{fe}(cba_k^-|dh_{sk}{}^+)$ is estimated as $\frac{1}{I+1}P_G(f|e'_*)$ or $\frac{1}{I+1}P_G(f|e''_*)$, here $|e| = 2$ and $I = 3$.

where the counts are computed over the fixed assignments $dh_{nk}{}^-$, and $P(e)$ is the relative frequency among all English words in the corpus.

The translation probability in the other direction is similarly computed as

$$P_{fe}(cba_k^+|dh_{sk}{}^-) : \left(\frac{1}{I+1}\right)^2 P_G(f'|e_*)P_G(f''|e_*), \tag{4.19}$$

where $P_G(f'|e_*)$ and $P_G(f''|e_*)$ are computed as

$$P_G(f|e) \quad = \quad (1-\alpha_2)\frac{N(f,e)}{N(e)} + \alpha_2 P_G(f). \tag{4.20}$$

In the second case, if the hypothesis in the evaluation does not have a word boundary at position k, the total number of Chinese words would be one less, i.e. J instead in the equations above, and there would be a single set of English words aligned to the word f in the E-to-C direction (see Figure **??**) and a single word e'_* or e''_* aligned to f in the C-to-E direction (see Figure **??**). The probability of this hypothesis is computed analogously.

The parameters θ are estimated on-the-fly, which means that updating θ indicates updating the counts $N(f,e)$, N, $N(e)$ and $N(f)$ according to the proposed model. The probabilities and counts are computed when they are called in the sampling.

4.6.6 Determining the set of alternative hypotheses

The sampling on the word segmentation can change the Chinese word and its alignment. Therefore, some implementation issues need to be addressed to enable the algorithm to work properly in the experiments.

4.6.6.1 How to maintain one-to-many alignment during sampling?

As mentioned earlier, we consider alternative alignments which deviate minimally from the current alignments and which satisfy the constraints of the IBM model 1 in both directions. In order to describe the set of alternatives, we consider two cases, depending on whether there is a boundary at the current character before sampling at position k.

Case 1. There was no boundary at c_k previously (see Figure **??** and **??**).

If there was no boundary at c_k, there is a single word f spanning that position. We denote by $\{e\}$ the set of English words aligned to f in the E-to-C direction and by e_* the English word aligned to f in the C-to-E direction in this case. Due to the fact that we consider the IBM model 1 one-to-many constraints, there is exactly one English word aligned to f in the C-to-E direction and the words $\{e\}$ have no other words aligned to them in the E-to-C direction.

In this case, we consider as hypothesis ${cba_k}^-$ the same segmentation and alignment as before. (see Table **??** for an overview of the alternative hypotheses.)

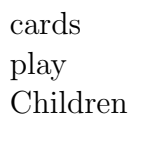
We consider $2^{|e|}$ different hypotheses which include a boundary at k in this case, where $2^{|e|}$ depends on the number of words aligned to f previously. As we are dividing the word f into two words f' and f'' by placing a boundary at c_k, we need to re-align the words $\{e\}$ to either f' or f''. Additionally, we need to align f' and f'' to English words in the C-to-E direction. These alternatives arise by considering that each of the words in $\{e\}$ needs to align to either f' or f'', and there are $2^{|e|}$ combinations of these alignments. For example, if $\{e\} = \{e_\prime, e_{\prime\prime}\}$ as in Figure **??**, after splitting the word f there are four possible alignments illustrated in Figure **??**: I. $(f', e_\prime)$ and $(f'', e_{\prime\prime})$, II. $(f', e_{\prime\prime})$ and $(f'', e_\prime)$, III. $(f', e_\prime)$ and $(f', e_\prime)$, IV. $(f'', e_{\prime\prime})$ and $(f'', e_{\prime\prime})$. For the alignment a_k in the C-to-E direction, we only consider one option, in which both resulting words f' and f'' align to e_*. These alternatives form ${cba_{k,e\_}}^+$ in Table **??**.

Case 2. There was a boundary at c_k previously (see Figure **??** and **??**).

In this case, for the hypothesis c_k^+ we only consider one alternative, which is exactly the same as the assignment of segmentation and alignments as previous.

Let f' and f'' denote the two words at position k previously. As in Figure **??**, e' and e'' denote the English words aligned to them in the E-to-C direction, respectively, and as in Figure **??**, ${e'}_*$ and ${e''}_*$ denote the English words aligned to f' and f'' in the C-to-E direction.

We only consider one hypothesis cba_k^- where there is no boundary at c_k. There is a single word f spanning position k in this hypothesis, and all words align to f in the E-to-C direction. For the C-to-E direction we approximately consider the 'better' one of the alignments (f, e'_*) and (f, e''_*) where the better alignment is defined as the one having higher probability according to the C-to-E word translation probabilities.



1	K_1 P(play|玩) P(cards|玩) P(玩) P(纸牌)
2	K_2 P(play|纸牌) P(cards|纸牌) P(玩) P(纸牌)
3	K_3 P(play|纸牌)P(cards| 玩) P(玩) P(纸牌)
4	K_4 P(play|玩)P(cards|纸牌) P(玩) P(纸牌)

Figure 4.7: An example of a word segmentation and alignment alternatives using Gibbs sampling.

For example, we need to decide on the boundary between 玩 and 纸牌. Figure **??** shows the alignment matrix of this sentence pair on the left side. The black boxes indicate the single-best alignments. We transit from no boundary to a boundary for E-to-C translation direction. That means previously 玩纸牌 is a single Chinese word aligning to 'play cards', and now we treat them as two words 玩 and 纸牌. Inserting a word boundary between 玩 and 纸牌 results in the re-alignment, therefore we receive four candidates. The computation of probabilities of these candidates is shown on the right side of Figure **??**, where K_1, K_2, K_3, K_4 are values for the length normalization.

4.6.7 Complete segmentation algorithm

So far, we have described how we re-sample word segmentation and alignments, starting from an initial segmentation and alignments from GIZA++. Putting these pieces together, we get the algorithm that is summarized in Table **??**.

Table 4.7: Complete algorithm of Gibbs sampler for CWS including alignment models. The observations are $D = (d_1, ..d_s, .., d_S)$, where d_s=(c_1^K, e_1^I) indicates a bilingual sentence pair. Hidden variables F_t and A_t indicate the word segmentation and word alignment of the corpus in the t-th iteration respectively.

```
1  Input: D, F_0
2  Output: A_T, F_T
3  for t = 1 to T : each iteration
4     Run GIZA++ on (D, F_{t-1}) to obtain A_t
5     Run GS on (D, F_{t-1}, A_t) to obtain F_t
```

We further improve performance by repeatedly aligning the corpus using GIZA++ for a more adequate re-alignment. We do so after deriving a new segmentation. The complete

algorithm which includes this step is shown in Table **??**, where F_t indicates the word segmentation at iteration t and A_t denotes the GIZA++ corpus alignment in both directions. The GS re-segmentation step is done according to the algorithm in Table **??**.

Using this algorithm we obtain a new segmentation of the Chinese data and train the translation models using this segmentation as in the baseline MT system. To segment the test data for translation, we use a unigram model trained with maximum likelihood estimation of the final segmentation of the training corpus F_T.

4.7 Integrated Chinese word segmentation in search

We described the learned and semi-supervised Chinese word segmentation methods in Section **??** and Section **??**, where the word segmentation is learned during the training of the word alignment. However, a test corpus is still processed using a unigram segmenter, which does not guarantee optimal segmentations as addressed in Section **??**. For instance, 小孩 and 孩 can both mean 'children'. The first one is used more often. Therefore, a segmenter usually puts both characters together rather than separating them. But if only 孩but not 小孩 appears in the training corpus, 小孩 should be broken into two words in the test corpus so that 孩 can be recognized and translated into 'children'.

Whenever inconsistencies appear between the expressions in the training and test data, it is a good idea to consider segmentation alternatives in order to adapt the writing style of the test text to that of the training text. Hence, a so-called 'integrated word segmentation' will be described in detail in this section as introduced in [**?**]. The algorithm works as follows: Given a set of character sequences as the input text, we take all possible segmentations of one sentence into account and integrate the segmentation decision into the search for the translation. Different segmentation possibilities represented as a lattice instead of a single segmentation are translated, and the segmentation decision is only taken during the search for the best translation.

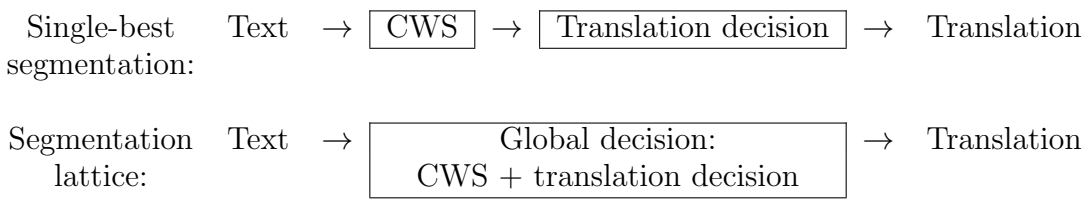


Figure 4.8: Translation procedures of the integrated Chinese word segmentation vs. single-best word segmentation.

Figure **??** shows the translation procedures of the integrated Chinese word segmentation based on lattices compared to a common translation procedure with single-best word segmentations as input. In the conventional method only the single-best word segmentation is employed into the search for the best translation. This approach is not ideal because the segmentation may not be optimal for these translations given the training data segmentation. Making hard decisions in word segmentation may lead to loosing Chinese words that can contribute to find the correct translations. Hence, for one input sentence, we take all possible segmentations into account and represent them as a lattice. The input to the translation system is then a set of lattices instead of the segmented text. As shown in Figure **??**, in the integrated segmentation the search decision of the word segmentation is combined with the translation decision as a global decision. The segmentation of a sentence is not selected until the translation is generated.

The following part of this section is structured as follows: First we will describe the model of the integrated word segmentation based on lattices in Section **??**. Then, the generation process of the segmentation lattices is described in Section **??** in detail. Finally we will show how to weight the segmentation alternatives using different feature costs in Section **??**, which improves the translation performance.

4.7.1 Integrated Chinese word segmentation model

In this section, we will explain the methods addressed in Figure **??** in detail. First, we will repeat a general word segmentation model described in Section **??**, which serves as a baseline and starting point to derive the integrated word segmentation model. Afterwards, we will present the integrated word segmentation model using the lattice translation.

In the description of common word segmentation models in Section **??**, a Chinese input sentence is denoted as c_1^K at the character level and f_1^J at the word level, where c_1, c_2,...,c_K are the succeeding characters and f_1, f_2,...,f_J are the succeeding words. $k_1^J = k_1 k_2 ... k_j ... k_J$ $(j \in 1, 2, .., J)$ are the segmentation boundaries. The optimal segmentation boundaries $\hat{k}_1^{\hat{J}}$ are obtained in the preprocessing step. $\hat{f}_1^{\hat{J}}$ indicates the word sequence, and its character sequence c_1^K is segmented by boundaries $\hat{k}_1^{\hat{J}}$, namely $\hat{f}_1^{\hat{J}} = c_1^{\hat{k}_1}, .., c_{\hat{k}_{j-1}+1}^{\hat{k}_j}, .., c_{\hat{k}_{\hat{J}-1}+1}^{K}$ and f_1^J is a word sequence of c_1^K segmented by boundaries k_1^J. The translation of c_1^K can be performed in two ways: **Single-best segmentation**

In the conventional approach only the best segmentation $\hat{k}_1^{\hat{J}}$ is translated into the target sentence:

$$\hat{e}_1^{\hat{I}} = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | c_1^K, \hat{k}_1^{\hat{J}}) \right\} \tag{4.21}$$

Segmentation lattice

In the transfer of the single-best segmentation from Equation **??** to Equation **??** some segmentations that are potentially optimal for the translation may be lost. Therefore, we combine the two steps. The search is then rewritten as:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | c_1^K) \right\} \tag{4.22}$$

We optimize the segmentation boundaries k_1^J to achieve the best translation directly. In this way, the segmentation model and the translation model are combined into one model. The global decision on Chinese word segmentation and translation are performed together.

4.7.2 Constructing segmentation lattices

To perform the lattice translation we introduce the weighted finite-state acceptor [**?**]. Now, we will take a short sentence as an example and simulate the segmentation process. The Chinese sentence in Table **??** was selected from the [**?**] development corpus. The sentence consists of nine characters and a punctuation mark: '在(at) 哪(what) 里(inner) 办(do) 理(manage) 登(ascend) 机(machine) 手(hand) 续(continue) ?'. After a manual

segmentation it contains six words: '在(at) 哪里(where) 办理(deal with) 登机(boarding) 手续(formality/formalities) ?'. Here, we use the manually segmented training corpus. As shown in Table **??** a translation on a lattice of different word segmentations leads to a better translation result.

Table 4.8: An example of simulating the process of the integrated Chinese word segmentation.

Source characters:	在	哪	里	办	理	登	机	手	续	?
c_1^K:	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
Manual source words:	在	哪里		办理		登机		手续		?
f_1^J:	f_1	f_2		f_3		f_4		f_5		f_6
Translation by										
single-best segmentation:	where to go through boarding formalities ?									
segmentation lattice:	where do I make my boarding arrangements ?									
One reference:	where do I complete boarding procedures ?									

There are many approaches which help to build a segmentation lattice. The aim of the lattice construction is on the one hand allowing word segmentation alternatives as candidates for translation and on the other hand avoiding too many ambiguities so that segmentations leading to optimal translations can always be preferred. We experimented on three types of segmentation lattices: single-best segmentation on words or characters, multiple segmentations generated by arbitrary segmentation methods as well as all segmentations given a word lexicon.

The simplest lattices are linearly constructed, i.e. a word sequence is taken as the only path in the lattice. In the case of infrequent ambiguities of the words in a sentence, a single-best segmentation on words can be applied.

1. Single-best segmentation on words

   Inside the translation systems the input sentence is represented in the form of a linear acceptor. Figure **??** shows the acceptor of the manually segmented sentence in Table **??**. Here, if any of the six words does not appear in the training corpus, then its translation would be missing. We note that not only the manual segmentation, but any word segmentation can be represented as a single-best segmentation such as the one performed by the unigram method in Section **??**.

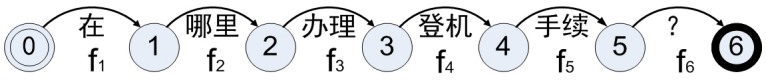


Figure 4.9: Single-best segmentation: the input sentence as a linear automaton

2. Single-best segmentation on characters

As described in Section **??** a straight-forward approach is to take each character as a single word relying on the phrase-based decoder to capture the context dependency among characters. This method is seldom applied in real applications because of suboptimal translation performance. However, plain translation on characters is quite often mentioned serving as a comparison and a baseline for refined segmentation methods.

3. Multiple segmentations

   In order to introduce segmentation alternatives N-best word segmentations instead of the single best segmentation are used in the translation. Chinese texts processed using different word segmentation methods are concatenated one after another to train the word alignment models. A segmentation lattice offers multiple paths with different segmentation possibilities allowing the decoder to take the final decision on the optimal word boundaries.

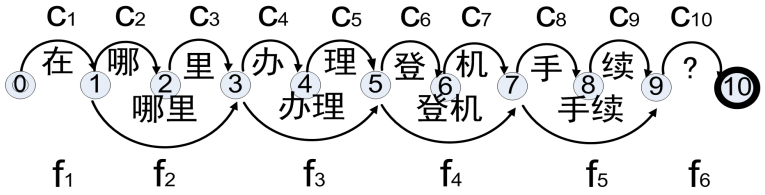


Figure 4.10: Segmentation lattice composed of a manual and a character-based segmentation

Figure **??** shows a segmentation lattice in the form of a finite state acceptor. The character-based segmentation and a manual segmentation '在 哪里 办理 登机 手续 ?' in Table **??** are combined in the lattice. The numbered states with one and eleven are the start and final state, respectively. Each arc is noted with its input label namely the corresponding Chinese word.

We can combine different segmentations in the lattice. For instance, we can add an automatic segmentation result '在 哪里 办理 登机手续 ?' to the lattice in Figure **??**, the acceptor in Figure **??** is extended where only the arc with '登机手续' is added between state five and state nine.

4. All segmentation alternatives

   If the vocabulary of Chinese words is given, it is possible to construct a lattice with all possible segmentations for a sentence. Allowing all alternative word segmentations tends to be a good idea, if several segmentations are not sufficient to detect words that are consistent with the training texts. This is realized by using the operator 'composition' within the framework of finite state acceptors introduced in the beginning of Section **??**.

   We generate the segmentation lattice using the following steps:

(a) First, we generate a word list shown in Table **??** from the vocabulary of the Chinese training corpus which contains all the entries that could be translated. Each word in the list is mapped to its characters to be consistent with the input of an unsegmented text. There may be several mapped words for one character sequence.

In order to avoid the problem of unknown characters from the unsegmented corpus, the additional characters from the test corpus are also added to the word list.

Table 4.9: A word list generated from the vocabulary of the Chinese training corpus.

characters	words
在	在
..	..
哪 里	哪里
办 理	办理
中 国 人	中国人

(b) We convert the mapping in Table **??** into a finite-state transducer for segmentation, as shown in Figure **??**. Here the input labels are the characters from the test corpus, and the output labels are Chinese words to be translated by the translation system. State 0 is the start and end state.

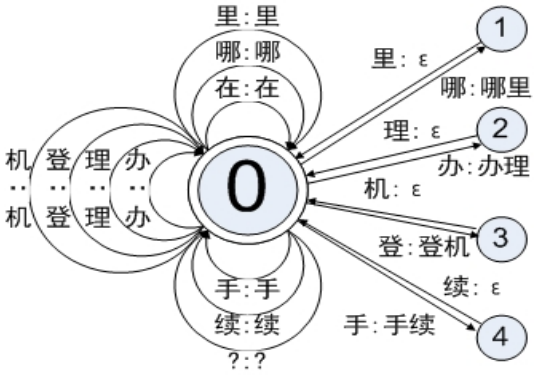


Figure 4.11: Segmentation transducer.

(c) The input character sequence is represented as a linear acceptor in the same way as the single-best segmentation, shown in Figure **??**.

(d) The linear automaton is composed with the segmentation transducer in Figure **??**. The result is a lattice which represents all possible segmentations of this sentence as shown in Figure **??**. Note that the alphabet in Figure **??** is a subset of the input alphabet in Figure **??** because the unknown characters are added as single words to the word list.

(e) Now, we get a new finite-state acceptor representing all alternatives of different word segmentations. We only need to read the segmentation lattice in

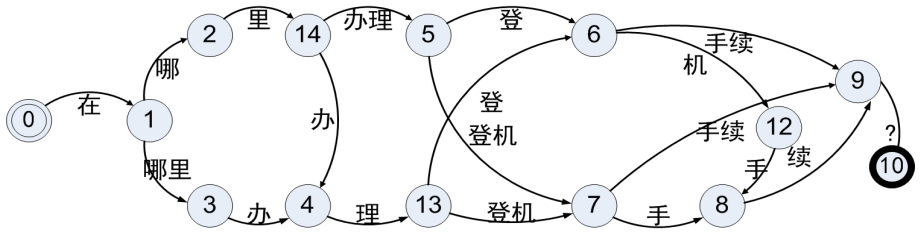


Figure 4.12: Segmentation lattice without weights including all word segmentation alternatives given a vocabulary.

Figure ?? instead of the linear acceptor in Figure ?? to have an integrated word segmentation in the translation.

4.7.3 Weighting segmentation lattices

The method for introducing segmentation alternatives is based on the assumption that the decoder is robust enough to choose the right segmentation using the translation model costs. However, if there are ambiguities, the decoder might prefer a path with lower translation costs without considering any context information. As a result, translations differ to a great extent from the original content. Therefore, we discuss possible features to evaluate different segmentations. Paths in a lattice are weighted by feature costs. Infrequent word segmentations are penalized and more frequent word segmentations gain priority. In this way features in the segmentation lattice and in the decoder both contribute to finding the best segmentation results.

We will describe two models to weight lattice paths. The first one is a length model based on the observation that single character words are often chosen without context meanings. The other one is a language model estimated on the Chinese training text. A unigram language model gives priority to frequent words used in the training data, and a higher order language model also captures the source context information for decisions.

- Length model

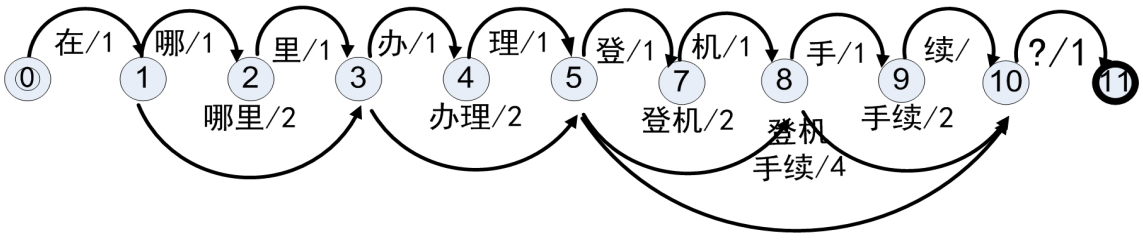


Figure 4.13: Three segmentations composed of a character-based segmentation, a manual segmentation and an automatic segmentation weighted by the word length model if $\eta = 1$.

The length model weight of one arc is estimated by the length of the word on the input label to the power of a parameter value η. For example, if the Chinese word on the input label contains two characters, then its weight is 2^{η}. The motivation for

this approach is to prefer longer words and to penalize single-character words. After applying the length model weights, each arc in the acceptor is assigned a weight as shown in Figure **??**.

- Language model

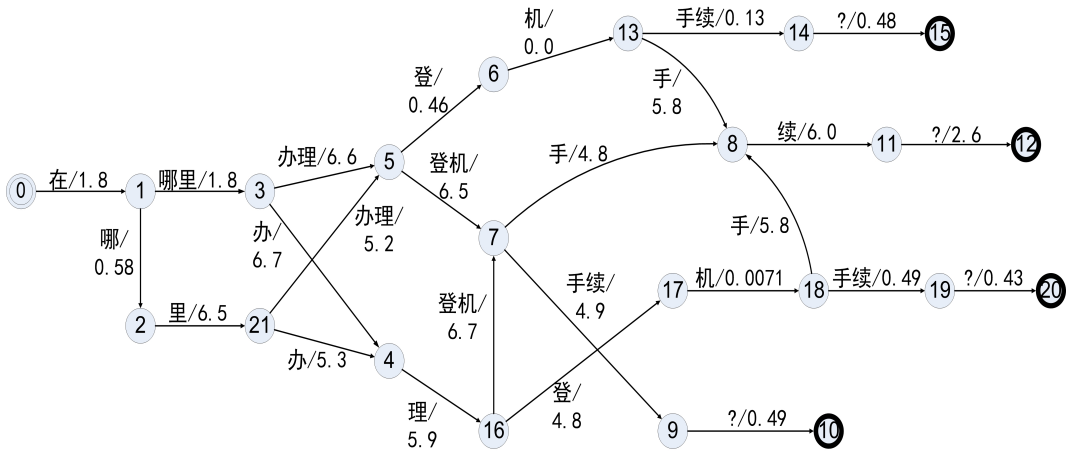


Figure 4.14: Segmentation lattice weighted by a language model considering all alternatives given a vocabulary.

A word segmentation model represents the fluency of a Chinese word sequence and can be built as an n-gram language model of the word-based text as formulated in Equation **??**. We trained the language model on the Chinese training corpus with the SRILM toolkit [**?**] and used the modified Kneser-Ney discounting. To combine the segmentation lattice with the word-based language model we simply transform the language model into a finite-state transducer and compose the lattice with it. Note that after inserting the weights the number of states and arcs in a lattice may increase because of differing language model histories.

Chapter 5

Phrase pair segmentation

Current statistical machine translation systems take phrases as units and perform translations based on phrase pairs. Inducing sufficient and accurate phrase entries is an elementary problem for a high quality translation system.

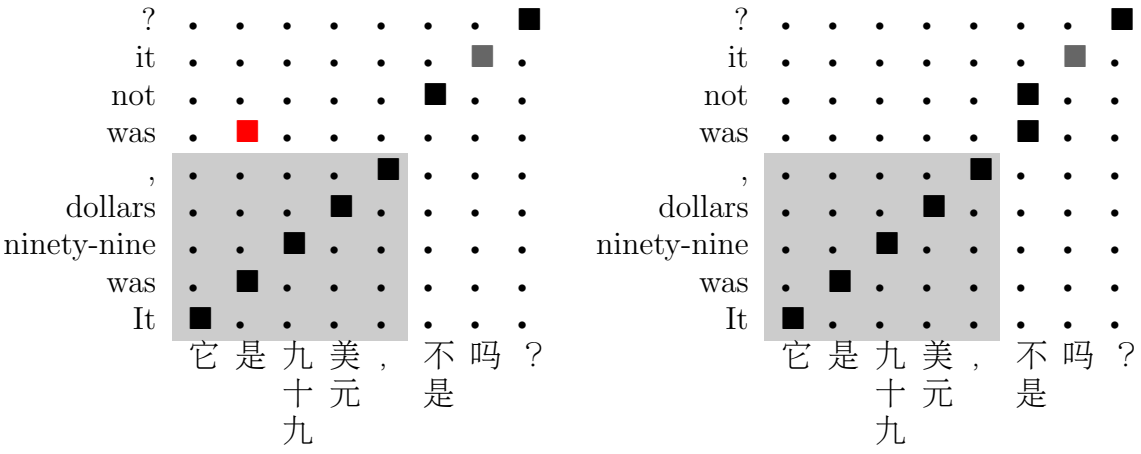


Figure 5.1: An example of phrase pair segmentation using the standard approach: the left figure shows that a wrong word alignment results in a missing phrase pair; the right figure shows that this missing phrase pair can be generated using the correct word alignment.

In the standard approach described in [**?**] and [**?**] the consecutive and consistently aligned words in the Viterbi word alignments are extracted as phrase pairs. This approach is widely applied but has two shortcomings: 1. As the extraction solely depends on the best word alignments a useful phrase pair could be missing if words are aligned incorrectly. Figure **??** shows an example of phrase pair extraction of a bilingual sentence selected from the [**?**] task. The Chinese words and their English glosses are '它(it) 是(is/was) 九十九(99) 美元(dollars) ， 不是(is/was not) 吗 ？' . There are two '是' in the source sentence and two 'was' in the target sentence. The first 'was' should align to the first '是', and the second 'was' is a translation of the second '是' which is a character in the word '不是' (was not). However, if the second 'was' misaligns to the first '是', the correct phrase pair '它是九十九美元， / It was ninety-nine dollars,' cannot be extracted in the standard approach. 2. Noisy phrase pairs can be contained without regard to linguistic

meanings. For example '不是吗 / not it' is extracted according to the algorithm in [**?**]. But 'not it' alone is syntactically incorrect, therefore errors may occur in the translation.

Therefore, we present an algorithm that allows to learn the phrase pairs discriminatively in order to maximize the overall translation performance and push learning down to the level of phrase extraction. All knowledge resources, such as probabilities derived from IBM model 1, HMM and other models, are treated as feature functions in the mixture model framework, which is easily to be extended by adding new feature functions. The standard phrase extraction is hence a special case by setting the feature weights of the other features to zero. Moreover, we introduced a bilingual entropy model to achieve better phrase pair precisions.

5.1 A mixture phrase model

We present a generic phrase training algorithm which is parametrized with feature functions and can be jointly optimized with the translation engine to maximize the end-to-end system performance directly. Multiple data-driven feature functions are proposed to capture the quality and confidence of phrase pairs. Experimental results demonstrate consistent and significant improvements over the widely used method that is only based on the word alignment matrices.

Now, we describe this approach in detail. A phrase table includes entries composed of a source phrase $\tilde{f} = f_{j_1}, .., f_{j_2}$ which is a sequence of words in the source language sentence starting from position j_1 and ending at position j_2, a target phrase $\tilde{e} = e_{i_1}, .., e_{i_2}$ which is a sequence of words in the target language sentence starting from position i_1 and ending at position i_2, and a score (cost) to evaluate how likely the phrase pair is irrelevant. Given a sentence pair f_1^J, e_1^I the cost of a phrase pair $(e_{i_1}^{i_2}, f_{j_1}^{j_2})$ is based on a mixture model combining several feature functions:

$$\sum_{m=1}^{M} \lambda_m h_m(i_1, i_2, j_1, j_2, f_1^J, e_1^I) \tag{5.1}$$

Each feature represents a phrase generation process indexed by m, where $1 \leq m \leq M$ is chosen randomly according to a feature weight λ_m indicating how likely the process m contributes to the final phrase pair score. Feature weights are discriminatively trained with the minimum error rate criterion. $h_m(i_1, i_2, j_1, j_2 | f_1^J, e_1^I)$ is the probability that $e_{i_1}^{i_2}$ is the translation of $f_{j_1}^{j_2}$ in the process m.

Figure **??** shows the basic architecture of a mixture model for phrase pair induction. The task is to classify a test source phrase $\tilde{f}$ into a target phrase $\tilde{e}$. The standard approach employs a single source of training data and one word alignment as input, while the mixture model is able to combine the phrases generated from multiple training domains and by different models.

The mixture model is trained in a way that each feature function h_m models a corresponding process and that the weight λ_m models the mixing parameter. Each feature

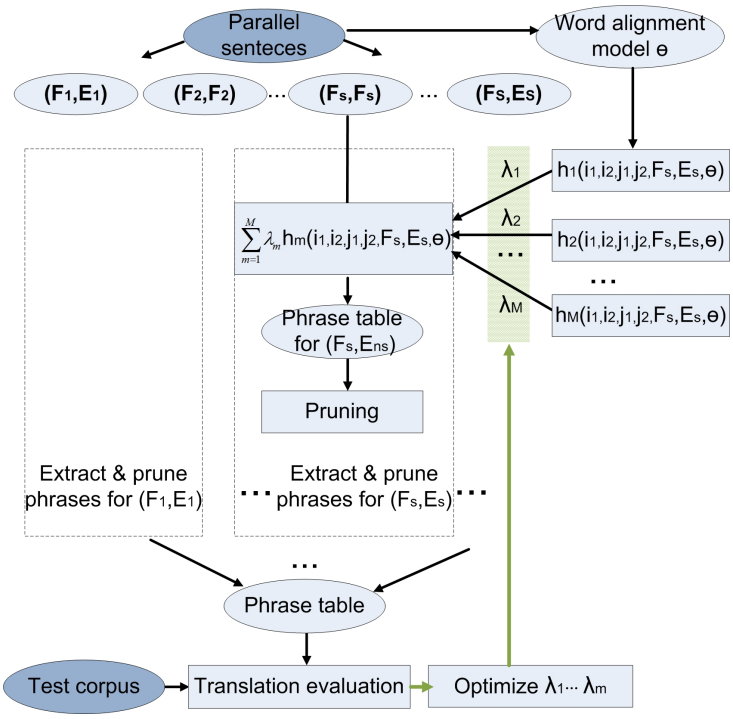


Figure 5.2: Architecture of mixture phrase table vs. standard phrase table generation.

function and its parameters can be obtained from different data resources and various models. We concentrate here on diverse model functions rather than on multiple data sets and therefore only perform experiments on the combination of features derived from several models on the same training data.

In Figure **??** the bilingual training data contains S parallel sentences $(F_1, E_1), (F_2, E_2), .., (F_s, E_s), .., (F_S, F_S)$, where (F_s, E_s) is the s-th sentence pair (F, E) also written as (f_1^J, e_1^I). We consider all possible phrase pairs $(\tilde{f}, \tilde{e})$ for each sentence pair (F_s, E_s) - in practice, we apply a phrase length restriction and compute M feature functions for each phrase pair given the sentence pair. The feature functions are derived based on the word alignment models and their parameters θ which are trained using GIZA++ [**?**] on the whole training set. A cost for a phrase pair can be assigned by summing all feature values h_m weighted by λ_m. We acquire a sentence level phrase table after accumulating all weighted phrase pairs for the sentence pair.

However, the complexity to generate all phrase pairs for a bilingual sentence is $O[\varphi_f \varphi_e (2J - \varphi_f)(2I - \varphi_e)]$, where J and I is the length of the source and target sentence respectively, and φ_f and φ_e are the maximum allowed phrase length of the source and target phrase respectively. Because of the high computational requirement, we introduce a threshold parameter to cut off phrase pairs with high costs. Phrase pairs assigned with a higher cost than the threshold are pruned. Merging the phrase tables generated from all sentence pairs, we obtain a global phrase table to perform the final translation on a test corpus. In order to receive an adequate phrase model, the mixture weights λ_1^M can be trained iteratively using the minimum error rate criterion. Therefore, both features and their weights which can also be viewed as a prior of the feature, are bound dynamically with the phrase model. Learning is pushed down to the level of phrase

extraction. All knowledge sources, such as probabilities derived by IBM model 1, HMM and other models, are treated as feature functions in the mixture model framework, which can easily be extended by adding new feature functions. The standard phrase extraction is hence a special case if the feature weights of other features are set to zero and the pruning step is omitted.

5.2 Phrase model features

Instead of only taking the Viterbi word alignment as input in the standard phrase extraction approach, we apply the underlying probability distribution derived directly from model training to compute feature costs. Now, we will present several feature functions that can be used in the mixture model of Equation **??**.

We will introduce data-driven features which are defined using the posterior distribution of the statistical word alignment models: One of them is based on the IBM model 1. The other one is based on the HMM word alignment model. The last one is based on the bilingual entropy to smooth the phrase boundaries.

It is assumed that a word alignment a in a statistical word alignment model indicates a target word e to be the translation of a source word f. Given a phrase pair in a sentence pair, there will be many paths (sequences of word alignments) to align the source phrase to the target phrase. The likelihood of those procedures can be accumulated to obtain the likelihood of the phrase pair [**?**], which is implemented as the summation of the likelihood function over all valid hidden word alignments.

5.2.1 IBM model 1 based on word posterior probabilities

$e_{i_2+1}, .., e_I$			
$e_{i_1}, .., e_{i_2}$			
$e_1, .., e_{i_1-1}$			
	$f_1, .., f_{j_1-1}$	$f_{j_1}, .., f_{j_2}$	$f_{j_2+1}, .., f_J$

Figure 5.3: Phrase extraction using IBM model 1 based on the posterior probabilities of word alignments. The posterior probability of a phrase pair alignment between $f_{j_1}, .., f_{j_2}$ and $e_{i_1}, .., e_{i_2}$ is defined as the sum of the posterior probabilities of the word alignments in the shaded areas.

The assignment of a probability to a phrase pair using IBM model 1 was investigated by [**?**] and [**?**]. Given a sentence pair (f_1^J, e_1^I), we want to find the translation of a sequence of words in one sentence in the other language. We can evaluate the relevance of a phrase pair as follows: We sum up word alignment posterior probabilities inside the target phrase for words inside the source phrase; and we sum up word alignment posterior probabilities outside the target phrase for words outside of the source phrase.

Let $A_{j_1,j_2}^{i_1,i_2}$ be the set of word alignments that aligns the source phrase $e_{j_1}^{i_1}$ to the target phrase $f_{j_2}^{j_1}$ (links to the *null* word are ignored for simplicity): $A_{j_1,j_2}^{i_1,i_2} = \{a : a_j \in$

$[i_1, i_2]$ iff $j \in [j_1, j_2]\}$. The alignment set given a phrase pair ignores those pairs with word links across the phrase boundary. Using IBM model 1, let $\gamma_\theta(i|j, f_1^J, e_1^I)$ be the posterior of $a_j = i$ given the sentence pair f_1^J and e_1^I and model parameters θ. The posterior of the alignment $A_{i_1,i_2}^{j_1,j_2}$ that is consistent with the phrase pair $f_{j_1}^{j_2}$ and $e_{i_1}^{i_2}$ is calculated as

$$P_1(A_{i_1,i_2}^{j_1,j_2}|f_1^J, e_1^I, \theta) = \prod_{j\in J_1}\sum_{i\in I_1}\gamma_\theta(i|j, f_1^J, e_1^I) \cdot \prod_{j\in J_2}\sum_{i\in I_2}\gamma_\theta(i|j, f_1^J, e_1^I), \tag{5.2}$$

where $J_2 = \{j_1, j_1 + 1, ..., j_2\}$ is the set of word indices of the concerned source phrase, $J_1 = \{1, 2, ..., j_1 - 1, j_2 + 1, ..., J\}$ the set of other source words, $I_2 = \{i_1, i_1 + 1, ..., i_2\}$ the set of word indices of the concerned target phrase, and $I_1 = \{1, 2, ..., i_1 - 1, i_2 + 1, ..., I\}$ the set of other target words. The left factor in Equation **??** relates to the alignments inside the phrase pair i.e. the gray area in Figure **??**. The right factor in Equation **??** relates to the alignments outside the phrase pair i.e. the light gray areas in Figure **??**. The feature obtained from the IBM model 1 posterior probability is defined as

$$h_1(i_1, i_2, j_1, j_2, f_1^J, e_1^I, \theta) = -\log P_1(A_{i_1,i_2}^{j_1,j_2}|f_1^J, e_1^I, \theta). \tag{5.3}$$

The phrase pair evaluation using the IBM model 1 posterior probability in the inverse direction $h_2(i_1, i_2, j_1, j_2|f_1^J, e_1^I, \theta)$ is calculated in the same way.

5.2.2 HMM based on word posterior probabilities

$e_{i_2+1}, .., e_I$			
$e_{i_1}, .., e_{i_2}$			
$e_1, .., e_{i_1-1}$			
	$f_1, .., f_{j_1-1}$	$f_{j_1}, .., f_{j_2}$	$f_{j_2+1}, .., f_J$

Figure 5.4: Phrase extraction using HMM model based on the posterior probabilities of word alignments. The posterior probability of a phrase pair alignment between $f_{j_1}, .., f_{j_2}$ and $e_{i_1}, .., e_{i_2}$ is defined as the sum of posterior probabilities of the word alignments in the shaded area after normalization.

Using HMM we evaluate the relevance of a phrase pair by adding up word alignment posterior probabilities inside the target and the source phrase with a normalization of the sum of all word alignment posterior probabilities in this sentence pair, as shown in Figure **??**. Consequently, the phrase pair posterior distribution based on HMM is defined as

$$P_3(A_{i_1,i_2}^{j_1,j_2}, f_1^J, e_1^I; \theta) = \frac{\sum_{a\in A_{i_1,i_2}^{j_1,j_2}}\gamma_\theta(a|f_1^J, e_1^I)}{\sum_{a\in A}\gamma_\theta(a|f_1^J, e_1^I)}, \tag{5.4}$$

where A is all alignments of a target word given a source word in this sentence (f_1^J, e_1^I), and $\gamma_\theta(a|f_1^J, e_1^I)$ can be efficiently calculated using the forward algorithm of HMM.

Equation **??** formulates the translation probability of a target to a source phrase. Switching source and target, we can obtain the posterior distribution in the other translation direction.

After transforming the probability into a cost, we get the feature function to represent the HMM posterior probability in the normal direction:

$$h_3(i_1, i_2, j_1, j_2, f_1^J, e_1^I, \theta) = -\log P_3(A_{i_1,i_2}^{j_1,j_2}|f_1^J, e_1^I, \theta) \tag{5.5}$$

The computation for the other direction $h_4(i_1, i_2, j_1, j_2, f_1^J, e_1^I, \theta)$ is analogous.

5.2.3 Bilingual entropy

We observed that some phrase pairs might have less meaningful phrase boundaries because the posterior probability presented before only shows how close the bilingual phrases are related to each other, but not how common a source or target phrase is used. Hence, we introduce a prior probability based on information entropy theory [**?**] to smooth phrase boundaries.

Here, the fewer target phrases a source phrase is aligned to, the more confident and convincing is it to represent the data and vice versa. The bilingual entropy of the source and target phrase can be calculated as in Equation **??** and in Equation **??** respectively:

$$\begin{aligned} & h_5(\cdot, \cdot, j_1, j_2, f_1^J, e_1^I, \theta) \\ &= -\sum_{1 \le {i_1}' \le {i_2}' \le I} P_3(A_{{i_1}',{i_2}'}^{j_1,j_2}|f_1^J, e_1^I, \theta) \log P_3(A_{{i_1}',{i_2}'}^{j_1,j_2}|f_1^J, e_1^I, \theta) \end{aligned} \tag{5.6}$$

$$\begin{aligned} & h_6(i_1, i_2, \cdot, \cdot, f_1^J, e_1^I, \theta) \\ &= -\sum_{1 \le {j_1}' \le {j_2}' \le I} P_3(A_{i_1,i_2}^{{j_1}',{j_2}'}|f_1^J, e_1^I, \theta) \log P_3(A_{i_1,i_2}^{{j_1}',{j_2}'}|f_1^J, e_1^I, \theta), \end{aligned} \tag{5.7}$$

where $1 \le {j_1}' \le {j_2}' \le J$ indicates all possible source phrase boundaries on the left and right side respectively in sentence f_1^J, and $1 \le {i_1}' \le {i_2}' \le I$ indicates all target phrase boundaries on the left and right side respectively in sentence e_1^I. The feature function is defined as the sum of the entropies in both languages.

5.3 Discriminative training

We would like to improve phrase translation accuracy and at the same time extract as many valid phrase pairs as possible that are missed due to incorrect word alignments.

Table 5.1: MER training for mixture feature weights in phrase pair segmentation.

1	Train model 1 and HMM word alignment models
2	Initialize an empty phrase table
3	for each sentence pair (f_1^J, e_1^I)
4	Identify candidate phrases on each side
5	for each candidate phrase pair $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$
6	Calculate its feature function values
7	Obtain the final score: $\sum_m \lambda_m h_m(i_1, i_2, j_1, j_2, f_1^J, e_1^I, \theta)$
8	Sort candidate phrase pairs by this final score
9	for each candidate phrase pair $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$
10	if $\left(\sum_m \lambda_m h_m(i_1, i_2, j_1, j_2, f_1^J, e_1^I, \theta) < \min_{i_1', i_2', j_1', j_2'} \sum_m \lambda_m h_m(i_1', i_2', j_1', j_2', f_1^J, e_1^I, \theta) + \tau\right)$ add this phrase pair into the phrase table
11	Use the phrase table to perform translations
12	Discriminatively train feature weights λ_1^M and threshold τ

We present a generic discriminative phrase pair extraction framework that can integrate multiple features aiming to identify correct phrase translation candidates. A significant deviation from most other approaches is that the framework is parametrized and can be optimized jointly with the decoder to maximize translation performance on a development set. We employ features based on word alignment models and alignment matrices. All of these features are data-driven and languages independent. The proposed phrase extraction framework generally applies to any other bilingual and monolingual feature as well as to linguistic features such as semantic and syntactic dependency.

Here, a minimum error rate (MER) training is employed to find optimal feature weights that maximize the final translation performance, which is achieved by minimizing the translation error rate on the development data. Previous successful MER applications in machine translation can be found for example in [**?**].

Table **??** shows the algorithm to train weights λ_m for feature $h_m(i_1, i_2, j_1, j_2, f_1^J, e_1^I, \theta)$ in a mixture phrase pair model discriminatively, where $m \in \{1, .., 6\}$. We calculate the translation result on a development corpus in each training iteration (line 1 to 11) given fixed feature weights and a threshold. Using the Powell algorithm [**?**], values of feature weights and pruning threshold that bring best translation performance are taken as the optimization results.

The process to perform translations based on a mixture phrase model is as follows: We first train IBM model 1 and HMM on the whole training corpus (line 1). Then for each sentence pair in the training corpus, we consider all phrase pairs that are shorter than a previously defined phrase length as candidate phrases (line 4). We introduced six different feature functions in Section **??**. These functions are evaluated for each phrase pair (line 6), and the function values are combined using feature weights (line 7). After that, all phrase pairs for a sentence pair are sorted according to their combined function value (line 8). Phrase pairs with a higher function value than the sum of the minimum function value and a threshold τ are pruned (line 10). Finally, translations are performed using the pruned

phrase table (line 11). This process is performed iteratively until the result converges.

This generic phrase extraction procedure is an evaluation, ranking, filtering, estimation and tuning process, and it can be described by the following steps:

1. Preparation of feature calculations

   Beginning with a uniform distributed lexicon we train IBM model 1 and HMM alignment models with 5 iterations for each translation direction using GIZA++ [**?**]. We use these models with parameters θ to evaluate candidate phrase pairs such as to calculate word alignment posterior probabilities as described in Section **??**.

2. Phrase pair selection

   This step consists of phrase pair evaluation, ranking and filtering. Each normalized feature score derived from the word alignment models will be combined to a final score. Phrase pair filtering is simply putting a threshold on the final score by comparing it to the minimum within the sentence pair.

3. Phrase pair evaluation

   This step pools all candidate phrase pairs that exceed the threshold test and estimates the final phrase translation table using the maximum likelihood criterion. To each candidate phrase pair which is below the threshold, we assign a model cost based on the phrase pair posterior probability and put the phrase pair into the global phrase table. One of the advantages of the proposed phrase training algorithm is that it is a parametrized procedure that can be optimized jointly with the translation engine to minimize the final translation errors measured by automatic metrics such as BLEU.

4. Feature weight optimization

   In the final step, parameters are trained discriminatively on a development set using the Powell method [**?**]. This phrase training procedure is configurable and trainable with different feature functions and their parameters. The commonly used phrase extraction approach based on word alignment heuristics as described in [**?**] and [**?**] is a special case of the algorithm where candidate phrase pairs are restricted to those which respect word alignment boundaries. We rely on multiple feature functions that aim to describe the quality of candidate phrase translations and the generic procedure to figure out the best way of combining these features.

Chapter 6

Sentence segmentation

We addressed two types of segmentation problems in machine translation and their solutions in Section **??** and **??**. Identifying proper boundaries for words and phrases are crucial issues in a state-of-the-art translation system. In Section **??** word boundaries are detected using Gibbs sampling and lattice translation; in Section **??** phrase segmentation and the alignment are performed under a mixture model with features based on posterior probabilities. In this chapter we will extend the sequence to be segmented and discuss the significance of segmentation on parallel sentences and paragraphs.

There are two major functions for sentence segmentation: efficient word alignment training and sentence alignment. In statistical machine translation word alignment models are trained on bilingual corpora. Long sentences pose severe problems: First, the high computational requirements, because the training is the most time-consuming part in the SMT process; Second, the poor quality of the resulting word alignment. We will present a sentence segmentation method that solves these problems by shortening long sentence pairs. Sentence pairs are split up using the so-called 'binary segmentation' method [**?**]. This algorithm leads to an improvement in translation quality and a significant speed-up of the training procedure. Furthermore, we apply the binary segmentation to the sentence alignment task in order to exploit parallel sentences effectively. Experimental results show an improvement in the translation performance over a state-of-the-art sentence aligner.

6.1 Binary segmentation

The main idea of the proposed sentence segmentation method is based on the so-called 'binary segmentation', i.e. we detect the optimal split point in a sentence pair and separate it into two pairs. The algorithm is inspired by the inversion transduction grammar (ITG) [**?**]. As the full parsing with ITG has a cubic complexity, it is too expensive to apply this algorithm on long sentences, which may be composed of over a hundred words. We approximately take the local decision after each recursion and present a top-down parsing concept to derive the best segmentation.

For a given sentence pair (f_1^J, e_1^I) or paragraph pair, each source position $j \in \{1, .., J\}$ in combination with each target position $i \in \{1, .., I\}$ is taken as a candidate segmentation

point.

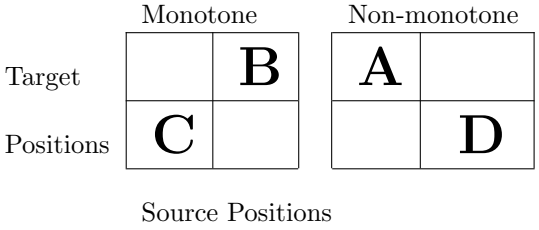


Figure 6.1: Two types of sentence alignment in binary sentence segmentation.

As shown in Figure **??** a candidate segmentation point (i, j) divides a word alignment matrix for this sentence pair or its subset (a segment pair) into four parts: the upper left (A), the upper right (B), the bottom left (C) and the bottom right (D) area. For a sentence pair with the start point $(1, 1)$ and end point (J, I) two types of alignments of sub-sentence pairs are possible in the binary segmentation:

1. Monotone alignment
   One case is the monotone alignment i.e. C is combined with B. We denote this case as $o = 1$ for alignment orientation. The segmentation cost is denoted as $h(j, i, 1, f_1^J, e_1^I)$.

2. Non-monotone alignment
   The other case is the non-monotone alignment indicated as $o = 0$. This means A is combined with D. We denote the cost as $h(j, i, 0, f_1^J, e_1^I)$.

All candidate segmentation points are computed with this method and the best splitting point and its orientation is selected so that the feature costs are minimized.

In most cases the sub-sentences, also called 'segments', are still too long after one splitting procedure. Therefore, the splitting is applied recursively until the length of each new segment is less than a predefined value. We introduce the maximum sentence lengths for the source language J_{max} and for the target language I_{max}. If one of the sentences in the pair is longer than the maximum length, the sentence pair is split into two segment pairs and each of the two segment pairs will be treated as a new sentence pair, and the same process will be iterated. Otherwise the sentence pair is kept unchanged and the segmentation process terminates.

Table **??** shows the recursive segmentation algorithm $\Upsilon(f_{j_1}^{j_2}, e_{i_1}^{i_2})$ for a bilingual sentence or segment $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$. The first and last word position in source sentence are j_1 and j_2, respectively; the first and last word position in the target sentence are i_1 and i_2, respectively. For sentence pairs that are longer than the user-defined maximum length, a best segmentation point and its orientation o is selected among all candidates according to Equation **??**. Two types of sentence alignments (orientations) are allowed as shown in Table **??**. If $\hat{o} = 1$, we process $(f_{j_1}^{\hat{j}}, e_{i_1}^{\hat{i}})$ and $(f_{\hat{j}+1}^{j_2}, e_{\hat{i}+1}^{i_2})$, otherwise $(f_{j_1}^{\hat{j}}, e_{\hat{i}+1}^{i_2})$ and $(f_{\hat{j}+1}^{j_2}, e_{i_1}^{\hat{i}})$ using the same algorithm.

Table 6.1: Recursive binary sentence segmentation procedure.

1	$\Upsilon(f_{j_1}^{j_2}, e_{i_1}^{i_2})$:	if	$(2 \leq j_2 - j_1 + 1 \leq J_{max}$ and $\leq i_2 - i_1 + 1 \leq I_{max})$
2		then	
3			$(f_{j_1}^{j_2}, e_{i_1}^{i_2})$: output the sub-sentence pair
4		else	
5			$(\hat{i}, \hat{j}, \hat{o}) = \underset{i,j,o}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(j, i, o, f_{j_1}^{j_2}, e_{i_1}^{i_2}) \right\}$,
6			where $i \in [i_1, i_2 - 1], j \in [j_1, j_2 - 1], o \in \{0, 1\}$
7			if $\hat{o} = 1$
8			then
9			$\Upsilon(f_{j_1}^{\hat{j}}, e_{i_1}^{\hat{i}}); \Upsilon(f_{\hat{j}+1}^{j_2}, e_{\hat{i}+1}^{i_2})$
10			else
11			$\Upsilon(f_{j_1}^{\hat{j}}, e_{\hat{i}+1}^{i_2}); \Upsilon(f_{\hat{j}+1}^{j_2}, e_{i_1}^{\hat{i}})$

6.2 Segmentation model

In order to include information from various resources, the sentence segmentation and alignment is evaluated by a mixture (log-linear) model combining different sub-models: the modified IBM model 1 in normal and inverse direction, the anchor words model as well as the IBM model 4.

Let (f_1^J, e_1^I) be a bilingual sentence to be split, the probability of a split point after (j, i) with orientation o is calculated using the following Equation:

$$(\hat{i}, \hat{j}, \hat{o}) = \underset{i,j,o}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(j, i, o, f_1^J, e_1^I) \right\}, \tag{6.1}$$

where $j \in [1, J-1]$ and $i \in [1, I-1]$ are positions in the source and target sentences respectively. To avoid the extraction of segments which are too short, e.g. single words, we use the minimum segment lengths (I_{min}, J_{min}). The possible split point is then limited to: $i \in [i_1 + I_{min} - 1, i_2 - I_{min}]$, $j \in [j_1 + J_{min} - 1, j_2 - J_{min}]$. M denotes the total number of different models. $h_m(j, i, o, f_1^J, e_1^I)$ is a score evaluated for (j, i) using sub-model m. Each model m is assigned with a feature weight λ_m. o is a Boolean variable to indicate the alignment monotonicity of the two sub-sentence pairs. The optimal split point and the alignment direction $(\hat{j}, \hat{i}, \hat{o})$ are found by traversing all positions of the sentence pair and maximizing the score, combining different features. The sub-models will be described in the following sections. The feature functions include

- h_1, h_2: normalized IBM model 1 in both directions
- h_3: anchor word model

- h_4: IBM model 4 word alignment

In most cases the sentence pairs are quite long, and even after one segmentation iteration we still may have long sub-segments. Therefore, we separate the sub-segment pairs recursively until the length of each new segment is less than a defined value.

6.2.1 Normalized IBM model 1

A shortcoming of the simple word-alignment-based model for the sentence segmentation is that the lengths of the separated sentence pairs are ignored. To balance the lengths of the two sub-sentence pairs, we normalize the alignment probability by the source sentence length and adjust its weight with a parameter β as described in [**?**]. Without considering empty words, IBM model 1 can be extended to:

$$p(f_1^J|e_1^I) = \prod_{j=1}^{J}\left(\frac{1}{I}\sum_{i=1}^{I}p(f_j|e_i)\right)^{\beta\cdot\frac{1}{J}+(1-\beta)} \tag{6.2}$$

The IBM model 1 for monotone alignment is therefore calculated as

$$h_1(j,i,1,f_1^J,e_1^I) = \log\left(p(f_1^j|e_1^i)^{\beta\cdot\frac{1}{j}+(1-\beta)}\cdot p(f_{j+1}^J|e_{i+1}^I)^{\beta\cdot\frac{1}{J-j}+(1-\beta)}\right), \tag{6.3}$$

and the non-monotone alignment is formulated in the same way:

$$h_1(j,i,0,f_1^J,e_1^I) = \log\left(p(f_1^j|e_{i+1}^I)^{\beta\cdot\frac{1}{j}+(1-\beta)}\cdot p(f_{j+1}^J|e_1^i)^{\beta\cdot\frac{1}{J-j}+(1-\beta)}\right) \tag{6.4}$$

The standard IBM model 1 calculates the conditional probability of a target sentence given a source sentence. The inverse IBM model 1 calculates the probability of a source sentence given a target sentence. By exchanging the source and target sentence or segment, the model using the inverse IBM model 1 is computed analogously.

6.2.2 Other features and alignment concatenation

- Anchor words

  Intuitively, some anchor words such as punctuation marks are more likely to be sentence boundaries. Preferring these anchor words as split points can effectively avoid the extraction of incomplete segment pairs. Therefore, we use an anchor word model to opt for the segmentations after special words, where the source and target words are identical. There are two options to realize this idea:

  1. Soft constraint

     Bonuses are assigned to those positions after anchor words when looking for split points. Segmentations are preferably placed after anchor words, but this is not a necessary condition:

$$h_3(j,i,o,f_1^J,e_1^I) = \begin{cases} 1 : f_j = e_i \wedge e_i \in \mathcal{A} \\ 0 : \text{otherwise} \end{cases} \tag{6.5}$$

$\mathcal{A}$ is a user defined anchor word list, here we use $\mathcal{A}$={.,"?;}. If the corresponding model scaling factor λ_3 is assigned a high value, the segmentation positions are placed most frequently after anchor words.

2. Hard constraint

   Segmentation boundaries are only allowed to take place after anchor words. This is a hard constraint and can be performed by looking for the split points at certain positions along the sentences. This means the search space in Equation **??** is limited to $i \in [I_{min}, I - I_{min}] \wedge j \in [J_{min}, J - J_{min}] \wedge f_j \in \mathcal{A} \wedge e_i \in \mathcal{A} \wedge f_j == e_i$.

- IBM model 4 word alignment

  If we already have the IBM model 4 Viterbi word alignments for the parallel sentences and need to retrain the system, for example to optimize the training parameters, we can include the Viterbi word alignments trained on the original corpora into the binary segmentation. In the monotone case the model is represented as

$$h_4(j,i,1,f_1^J,e_1^I) = \log\left(\frac{N_a(f_1^j,e_1^i) + N_a(f_{j+1}^J,e_{i+1}^I)}{N_a(f_1^J,e_1^I)}\right), \tag{6.6}$$

  where $N_a(f_1^j,e_1^i)$ denotes the number of alignment links inside the matrix $(1,1)$ and (j,i). In the non-monotone case the model is formulated in the same way.

- Word alignment concatenation

  In phrase-based translation, we extract all phrases matched in the training corpus for an input sentence and translate with these phrase pairs. During sentence segmentation we might separate a phrase into two segments so that the whole phrase pair can not be extracted.

  To avoid this, we concatenate the word alignments trained with segment pairs extracted from one sentence pair in their original order. During the segmentation, the position of each segmentation point in the sentence is memorized. After training the word alignment model with the segmented sentence pairs, the word alignments are concatenated again according to the positions of their segments in the sentences. Finally the original sentence pairs and the concatenated alignments are used for the phrase extraction.

6.2.3 Efficient IBM model 1 computation

Among all above mentioned features the IBM model 1 and its inverse direction needs most of the computation time. The naive implementation of this algorithm using IBM model 1 has a complexity of $O((I \cdot J)^2)$. We benefit from the structure of the IBM model 1 and calculate the alignment probability for each position using the idea of 'running sums/products'. The complexity is reduced to $O(I \cdot J)$ which is factor of 10 000 for sentences with 100 words. But this implementation is not possible for the fertility-based higher-order models.

Table 6.2: Efficient computation of IBM model 1.

```
1   Max = 0;
2   ∀j ∈ [j1, j2] : Vup[j] = Σ_{i=i1}^{i2} p(fj|ei);
3   ∀j ∈ [j1, j2] : Vdown[j] = 0;

4   for   (i = i1; i < i2; i = i + 1)

5         ∀_{j∈[j1,j2]} : Vup[j] = Vup[j] − p(fj|ei);
6         ∀_{j∈[j1,j2]} : Vdown[j] = Vdown[j] + p(fj|ei);

7         A = C = 1;
8         B = Π_{j=j1}^{j2} Vup[j];
9         D = Π_{j=j1}^{j2} Vdown[j];

10        for   (j = j1; j < j2; j = j + 1)

11              A = A · Vup[j];
12              B = B/Vup[j];
13              C = C · Vdown[j];
14              D = D/Vdown[j];

15              if    (max(A · D, B · C) > Max∧
16                    i ∈ [i1 + Imin − 1, i2 − Imin]∧
17                    j ∈ [j1 + Jmin − 1, j2 − Jmin])
18              then
19                    Max = max(A · D, B · C);
20                    ĵ = j; î = i;
21                    m̂ = (B · C >= A · D);
```

Details are shown in Table **??**. The input to the program is the lexicon probabilities $p(f_j|e_i)$ and the minimum sentence lengths J_{min}, I_{min}. The output is the optimal split point $(\hat{j}, \hat{i})$ and its orientation $\hat{o}$.

In the program Max is the highest alignment probability. A, B, C and D are the IBM model 1 scores for each block in Figure **??**. V_{up} stores the sums of the lexicon probabilities in each column in the areas A and B, and V_{down} does the same for the areas C and D.

In the outer loop of the target position i, $p(f_j|e_i)$ in the actual position is added to and subtracted from the value in V_{down} and V_{up}, respectively. In the inner loop of the source

position j, the alignment probability in the area A/B are multiplied/divided by $V_{up}[j]$, and the probability in C/D is multiplied/divided by the $V_{down}[j]$. After traversing all positions the point with the maximum alignment probability is selected as the split point.

6.2.4 Segmentation example

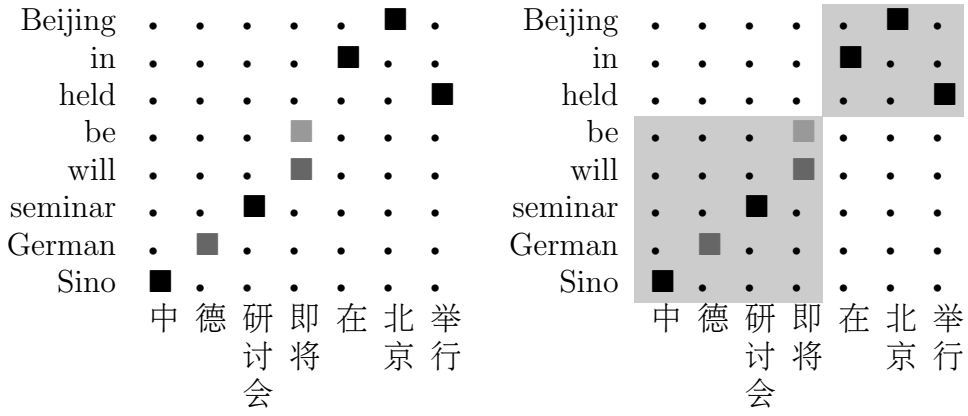


Figure 6.2: Word alignment matrix of a sentence pair, where darker blocks indicate a lexicon probability. The shaded area indicates the alignments of the sub-sentence pairs after the first iteration of segmentation.

Now, we will provide an example to explain the segmentation procedure. The Chinese sentence in words is '中(China) 德(German) 研讨会(seminar) 即将(will) 在(in/at) 北京(Beijing) 举行(held/hold)'. As illustrated in Figure **??**, we present the word alignment of this Chinese sentence and its English translation as a matrix. Each position contains a lexicon probability $p(f_j|e_i)$ which is trained on the original bilingual corpus, where long sentences are truncated to a maximum size, e.g. one hundred. For a clearer presentation, Figure **??** shows a short sentence pair of seven Chinese and eight English words. The gray scale indicates the value of the probability. The darker the box, the higher the word alignment probability. All positions are considered as possible split points. GB

<table>
<tr><td colspan="6">中德 研讨会 即将 在 北京 举行 / Sino German seminar will be held in Beijing</td></tr>
<tr><td colspan="3">中德 研讨会 即将 / Sino German seminar will be</td><td colspan="3">在 北京 举行 / held in Beijing</td></tr>
<tr><td colspan="2">中德 研讨会 / Sino German seminar</td><td>即将 / will be</td><td colspan="2">在 北京 / in Beijing</td><td>举行 / held</td></tr>
<tr><td>中德 / Sino German</td><td>研讨会 / seminar</td><td>即将 / will be</td><td>在 / in</td><td>北京 / Beijing</td><td>举行 / held</td></tr>
</table>

Figure 6.3: Result of the sentence segmentation example.

Using the algorithm of Table **??**, the sentence pair is segmented as in Figure **??** if we set the maximum sentence length in both languages to one. First, the lengths of the two sentences are longer than the maximum lengths, thus sentences will be segmented. After the calculation with Equation **??** we obtain the first segmentation point between '即将' and '在' in the source language and between 'be' and 'held' in the target language i.e. $\hat{j} = 4, \hat{i} = 5$. The alignment is monotone i.e. $\hat{o} = 1$. The Chinese sentence is segmented

into two parts '中德研讨会即将' and '在北京举行', and the English sentence is segmented into "'Sino German seminar will be"' and "'held in Beijing"'. After the first iteration, both segments in Chinese should be further separated. Hence, the first segment pair is split up again into two pairs '中德研讨会/Sino German seminar' and '即将/will be' with a monotone alignment, and the second segment pair is split up into '在北京/in Beijing' and '举行/held' with a non-monotone alignment. The recursion stops when each segment contains a single word. Note that in real applications the maximum allowed segment length is usually longer than twenty.

6.3 Bitext exploitation

In statistical machine translation, a large number of parallel sentences are required to train the model parameters. However, plenty of bilingual language resources that are available on web are only aligned at the document level. To exploit this data, we have to extract the bilingual sentences from these documents.

The common method is to split the documents into sentences using predefined anchor words such as punctuation marks and then to align these sentences. This is the so-called 'sentence alignment task'. However, it could be sub-optimal for the translation task if we only assume positions at the anchor words as sentence boundaries, and incorrect alignments may also decrease the translation quality.

We employ the sentence segmentation model in Section **??** for the sentence alignment task and combine it with the approach of [**?**]. The corpora produced using both approaches are concatenated, and a weight is assigned to each corpus. During the training of the word alignment models, the counts of the lexicon entries are linearly interpolated using the corpus weights. We will describe the different methods to extract the bilingual sentence pairs from the document aligned corpus in detail.

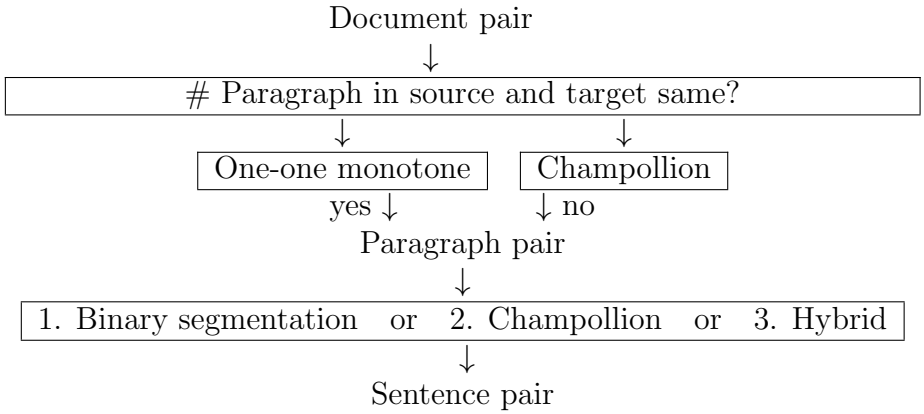


Figure 6.4: Sentence alignment using binary sentence segmentation method, dynamic programming algorithm (Champollion) and a hybrid approach.

As shown in Figure **??**, given each document pair, we assume that the paragraphs are aligned monotonically one to one if both the source and target language documents

contain the same number of paragraphs. Otherwise, the paragraphs are aligned with the Champollion tool. After obtaining the paragraph aligned corpus, we can use binary segmentation as described in Section **??**, dynamic programming e.g. Champollion or a Hybrid approach to align bilingual sentences:

1. Binary segmentation

   The segmentation method described in Section **??** is applied by treating the paragraph pairs as long sentence pairs. Segmentation of paragraph pairs into sentence pairs is realized by segmentation of long sentence pairs into shorter segment pairs. We can use the anchor words model described in Section **??** to prefer splitting up at punctuation marks.

   The lexicon parameters $p(f|e)$ in Equation **??** are estimated as follows: First, the sentences are aligned roughly using the dynamic programming algorithm. We get the initial lexicon parameters while training on these aligned sentences. Then the sentence segmentation algorithm is applied to extract the sentences again.

2. Champollion

   After paragraphs are divided into sentences at punctuation marks in both languages, the Champollion tool [**?**] is used to apply dynamic programming based sentence alignment.

3. Hybrid approach

   The binary segmentation and the Champollion method search for alignments using different approaches: the previous one is based on the ITG, and the latter one is based on the dynamic programming. Therefore, we combine these two methods by concatenating the corpora produced by both methods and using them in the word alignment training. A weight is assigned to each corpus. That means, the counts of the lexicon entries in the EM algorithm are linearly interpolated using the corpus weights.

Chapter 7

Document segmentation

While statistical machine translation systems have been improved significantly with better modeling techniques and an increasing amount of training data, domain specific SMT has received much less attention and leaves much room for further improvements. For instance, the documents to be translated can have different language styles. The language style of broadcast conversation is very different from that of the newswire text.

In this chapter, we will present domain dependent machine translation with a series of problems to be solved: The first one is how to partition (cluster) training documents into multiple domains; The second one is how to build domain specific SMT systems in training and obtain the prior probability of each domain given a development set; The last one is how to perform domain adaptation during decoding.

In this chapter, we address an unsupervised document clustering method to segment the corpus into multiple parts depending on their domains, which makes it easy and efficient to capture as many domains as required. For the second problem we use domain dependent language modeling and model combination. Domain priors are optimized with respect to the translation performance on a development set. Finally, domain adaptation during decoding is approached using source text classification methods. When translating a test document we will identify its domain automatically and then apply a corresponding decoding setup.

7.1 Document clustering

Training corpora delivered by linguistic organizations are in general sets of documents collected from diverse resources. They can be newswire text, broadcast transcriptions, Internet conversations or any other type. A newswire text usually has longer sentences in correct grammar, while the sentences in a conversation record may be short and formulated simpler. Another instance is that a scientific article is expected to contain more words like 'distribution' or 'problem', while in a document from travel agency words like 'flight' and 'restaurant' appear more often. A translation system trained on a travel domain might fail in translating a scientific document and vice versa. Therefore, properly classifying documents into multiple domains properly turns out to be a crucial issue for translating

a certain type of test data.

Ideally, we would have domain specific training data and could build a separate SMT system for each domain. Practically, this is hardly the case. We assume that we have a collection of training corpora from the general domain including a variety of different domains. At the same time we have a small amount of domain specific parallel documents to be used for building the domain specific systems.

We focus on performing clustering to achieve the best translation performance. In theory, the clustering problem can be solved by exhaustive enumeration. However, there are $\frac{c^n}{c!}$ ways to partition n documents into c clusters, and this exponential growth with n is overwhelming and difficult to implement. Without an initialization, we can intuitively follow either a top-down approach (all documents are regarded to be in one cluster, and the clusters are split iteratively) or a bottom-up approach (each document forms its own cluster, the clusters are merged step by step). In the experiments, we take the second approach because we only need to compute the similarity between clusters approximately.

The bottom-up method was described in [**?**] and applied by [**?**] for speech recognition. In selecting which clusters should be merged, we take two kinds of information into account: the number of samples in each cluster and the similarity between clusters. In general, this method tends to favor growth by merging singletons or small clusters with large clusters over merging medium-sized clusters. Let W_r be the set of unique content words in cluster r, $|W_r|$ be the number of words in W_r, and $|\mathcal{D}_r|$ be the number of documents in cluster r, then the similarity measure between cluster r_1 and cluster r_2 is defined as

$$\mathcal{S}(\mathcal{D}_{r_1}, \mathcal{D}_{r_2}) = \sqrt{\frac{|\mathcal{D}_{r_1}| + |\mathcal{D}_{r_2}|}{|\mathcal{D}_{r_1}||\mathcal{D}_{r_2}|}} \frac{|W_{r_1} \cap W_{r_2}|}{|W_{r_1} \cup W_{r_2}|}. \tag{7.1}$$

The similarity is proportional to the number of common words in cluster r_1 and r_2 but is anti-proportional to the number of all words in both clusters. The left factor in Equation **??** serves as a normalization factor to avoid the tendency for small clusters to group with one large cluster rather than other small clusters.

Table 7.1: Stepwise optimal hierarchical clustering using bottom-up method (agglomerative hierarchical clustering) where $\mathcal{D}_r$ indicates the r-th domain ($r \in 1, .., R$), and u_q indicates q-th document ($q \in 1, .., Q$).

1	Initialize: each document has its own cluster: $R = Q$ and $\mathcal{D}_r = \{u_q\}$
2	while (!terminate)
3	$\mathcal{D}'^{R'}_1 = \mathcal{D}^R_1$ and $R' = R$
4	$(\hat{r_1}, \hat{r_2}) = \text{argmax}_{r_1 \neq r_2} \mathcal{S}(\mathcal{D}_{r_1}, \mathcal{D}_{r_2})$: find two clusters most similar
5	merge $\mathcal{D}_{\hat{r_1}}$ and $\mathcal{D}_{\hat{r_2}}$
6	update $\mathcal{D}^R_1$ and decrease R with one
7	if ($\psi(\mathcal{D}^R_1, \mathcal{D}'^{R'}_1)$) terminate
8	Output R as optimal cluster number and D^R_1 as optimal document clustering

Once the similarity measure between two clusters is defined, we can apply the agglomerative clustering as shown in Table **??**. Given a number of documents $u_1, u_2, .., u_q, .., u_Q$, we initialize each document as its own cluster where $\mathcal{D}_r$ only contains one document u_r and the number of clusters is equal to the number of documents. In each iteration the similarity of any two distinct clusters is calculated. The two clusters with the highest similarity are merged into one. The number of clusters is then decreased by one. This process is performed iteratively until the termination condition is satisfied. The output is the clusters D_1^R in the last iteration and the number of these clusters R.

The termination condition $\psi(\mathcal{D}_1^R, \mathcal{D'}_1^{R'})$ can be calculated in many ways, such as the previously described in [**?**] and [**?**]. Here, we can heuristically check whether the degradation of the maximaum similarity between any of two clusters is smaller than a given value. While ideally, in order to decide on which level of the hierarchy the action of merging clusters stops, we should measure the translation performance with respect to a criterion whenever a new cluster is generated. We compare translation performances based on the clustering results in all iterations and choose the clustering that leads to the best translation performance. However, this approach is limited by a high complexity. Let Q be the document number, then in the worst case $Q - 1$ translation operations are required. As the corpus might contain millions of documents, it is a difficult task to perform such frequent translations.

Nonetheless, we observed an anti-proportional relationship between the perplexity of the language model in search and the final translation performance in the experiments, which means that we only need to minimize the perplexity of the language model instead of maximizing the final translation quality. Therefore, the termination constraint $\psi(\mathcal{D}_1^I, \mathcal{D'}_1^{I'})$ can be defined as the difference of two perplexities on a given test set. These two perplexities are measured using the mixture language models trained on the data which are clustered in the previous and in the current iteration respectively. The program terminates if the difference is smaller than a user defined threshold. This method avoids the drawing of the whole clustering hierarchy but terminates when an acceptable clustering has been achieved with respect to the language model quality.

7.2 Building domain specific translation systems

We described methods to cluster monolingual training data into appropriate domains based on the unigram coverage in Section **??**. Now, we will employ this data and discuss how to build domain specific SMT systems on it. We avoid building separate phrase translation tables for various domains due to computational requirements, and only one general phrase table is shared among all domain specific systems. Domain dependent translation systems are built under two approaches: adapted language models and domain specific translation model combination.

1. **Domain dependent language modeling**

   We apply a general decoder but use different domain dependent language models. Using the method in Section **??**, documents can be clustered according to their contents automatically. As a result, we have training data on the target side in R

domains. A hypothesis is then evaluated by a language model that is a mixture of R sub-language models [?]. Each sub-language model indexed with r is trained on domain-specific data cluster r. Here, we overload notations and use J to denote the number of words in a test corpus. Let $P(f_j|\cdot, r)$ be the probability of the j-th word in the test corpus estimated by the r-th sub-language model. The probability of a word sequence $f_1, .., f_J$ is modeled as

$$P(f_1, .., f_J) = \prod_{j=1}^{J} \sum_{r=1}^{R} \lambda_r P(f_j|f_{j-n+1}^{j-1}, r), \tag{7.2}$$

where the weight λ_r is the prior probability of the r-th sub-language model. Equation **??** defines a mixture language model linearly combining probabilities estimated by domain-specific n-gram language models. The feature weights λ will be optimized with respect to a lower perplexity tested on a predefined development corpus. We will discuss the optimization methods in Section **??**.

A mixture language model has the strength to predict the word distributions in the right domain. For instance, if the content of a test set is about traveling, we will be able to enhance the weights of the sub-language models of this domain automatically so that the writing style and the n-gram coverage of the translation is closer to that of a text in the tourism-related domain.

Moreover, training language models separately reduces the computational requirement significantly. We observed that increasing the amount of training corpus could help to enhance the language model quality. However, the size of the language model is limited by the computational resources because the training of a language model on huge corpora requires a large memory. Therefore, we train sub-language models on each cluster respectively instead of training a universal language model on the above data. Experiments in Section **??** show that with the same memory limit, we are able to lower the language model perplexity by about 15% absolutely using this approach, which leads to a significant improvement in translation performance.

2. **Domain dependent model combination**

   Another method to build a domain specific SMT system is to configure the feature weights in the log-linear model combination of the decoder i.e. scaling factors of phrase relative frequency model, language model, word penalty model and so on. Details of these models can be found in [?]. Domain specific scaling factors are trained discriminatively on the domain specific development set. For instance, a sentence in a blog may be shorter than that in a news article, where a higher word penalty is preferred.

The major advantage of both of the methods discussed above is that we only need a small amount of bilingual data from each domain, which is much easier to obtain. The monolingual data for each domain is clustered by the method in Section **??**.

7.3 Extremum of functions

We mentioned the optimization problem of feature weights in Section **??**, more precisely, the estimation of the λ_r in Equation **??**. The optimization problem is defined in [**?**]: Given a single function that depends on one or more independent variables, we want to find the value of those variables where the function takes on a maximum or a minimum value. Performing this optimization fast and memory-efficient is a key issue.

In translation tasks the optimization problem appears frequently. For example, feature weights in the log-linear model combination of the decoder are optimized with respect of the translation performance. Assigning appropriate feature weights can enhance the translation performance up to 20% relatively measured in the BLEU score. The other instances can be found in the log-linear combination of the semi-supervised Chinese word segmentation (Equation **??**), of the phrase models (Equation **??**) and of the sentence segmentation (Equation **??**). Therefore, the maximization or minimization of functions is of crucial importance in building an efficient translation system. In this section we will only consider the example of optimizing the mixture language model weights (Equation **??**). Nevertheless, the algorithms can be applied to the other problems as well.

A domain dependent language model formulated in Equation **??** is a mixture of several sub-language models trained on different domains. The vector λ_1^R measures how sub-language models contribute to the mixture language model, which contains the variables to be optimized. We take the language model perplexity measured on a development corpus as the evaluation criterion. As we need to optimize a set of feature weights together, the optimization algorithm has to suitable for the multidimensional case. We propose two algorithms, the *downhill simplex* method and the *Powell's method*, where only function evaluations but not derivatives are required.

The downhill simplex method was developed by [**?**] and is slow but extremely robust, with a memory requirement in the order of R^2. A simplex is the geometrical figure consisting of $R+1$ points in R dimensions and all related lines and faces, etc. In a two-dimensional space, a simplex is a triangle. If any point of the simplex is taken as the origin, then the R other points define vector directions that span the R dimensional vector space. The algorithm starts with a random, R-vector of independent variables as an initialization, then continues with a series of steps updating the simplex to get close to the function minimum, and finally the calculation terminates if the decrease of the function value is smaller than a given tolerance threshold. A detailed description can be found in [**?**].

Powell's method is a prototype of 'direction-set' methods with a storage of order R^2. Powell's method is faster than downhill simplex. The main idea is as follows: There is always a starting point and a vector of R independent directions for search. We start at a point in the R-dimensional space and proceed from there to one-dimensional optimization sequentially in R directions. A new direction is decided using this starting point and the endpoint after R one-dimensional optimizations. We replace one of the directions in the vector with this new direction, then a new direction vector is created. The new starting point is set as the minimum along the new direction starting from that end point. This process is iterated until the extremum is found.

Downhill simplex and Powell methods can be applied to estimate scaling factors of log-

linear models in decoding, feature weights for semi-supervised Chinese word segmentation as well as prior probabilities of sub-language models in a mixture model. Here, for the last case we also introduce an EM algorithm to find the most suitable λ in Equation **??**.

Table 7.2: EM algorithm of feature weights λ optimization in a unigram mixture language model. $\lambda_r \in \{1, .., R\}$: the weight of the r-domain; $j \in \{1, .., J\}$: a word position in the development corpus; σ: a user-defined threshold.

1	Input: initialization λ_1^R, $t = 0$
2	Output: final estimated $\hat{\lambda}_1^R$
3	for each pair (j, r)
4	calculate $P(f_j\|r)$
5	while (!terminate)
6	for each domain r
7	$\hat{\lambda}_r = \frac{1}{J} \sum_j \frac{\lambda_r P(f_j\|r)}{\sum_{r'} \lambda_{r'} P(f_j\|r)}$ (Equation **??**)
8	if ($\|\hat{\lambda}_r - \lambda_r\| < \sigma$) terminate
9	$\lambda_r = \hat{\lambda}_r$

We experiment with a method provided by the SRI tool kit [**?**] and the feature weights of the sub-language models λ is updated as

$$\tilde{\lambda_r} = \frac{1}{J} \sum_{j=1}^{J} \frac{\lambda_r P(f_j|r)}{\sum_{r'=1}^{R} \lambda_{r'} P(f_j|r')}, \tag{7.3}$$

where $r = 1, .., R$ is the index of a sub-language model to be merged, and $j = 1, .., J$ is the position of a word in the test set. All sub-language models should be evaluated on the same development corpus. $P(f_j|r)$ is the probability of the j-th word estimated by the r-th sub-language model. For simplicity, we only consider training sub-model weights for a unigram mixture language model. However, this algorithm can be generalized for any higher-order language model.

Table **??** represents the Expectation Maximization (EM) algorithm for the optimization. The initialization consists of a vector of weights λ_1^R and a matrix of probabilities $P(f_j|r)$ for each word on position j in the development corpus estimated by the r-th sub-language model. The output is the estimated vector of $\hat{\lambda}_1^R$ which fills the constraint that the change of posterior probability for each domain is smaller than a user defined value σ (line 8). The posterior probabilities estimated by the r-th language model are accumulated over all words (line 11) and normalized to compute the new estimation of λ_r for an update (line 5-9).

7.4 Domain adaptation

Before decoding a test document, we decide which domain specific SMT system we apply. Therefore, the domain adaptation is transformed into a monolingual text classification problem: Which domain is the test document most similar to?

In theory any text classification method can be applied here. We investigate two text classification techniques: one based on domain specific language models and the other based on information retrieval techniques.

7.4.1 Language model based domain identification

Here, we will consider two domains as an example: newsgroup and newswire. We build domain specific language models P_d ($d \in \{1, 2\}$) with the source side of the development sets for each domain. Note that we need to distinguish these from the domain specific language models trained on the target side of the corpus described in Section **??**. As the development sets are usually small, each of the models P_d is linearly interpolated with a general domain independent language model P_g:

$$P_d^*(f|\cdot) = (1 - \alpha)P_d(f|\cdot) + \alpha P_g(f|\cdot) \tag{7.4}$$

For a test document to be translated, we compute the perplexity of each domain specific language model P_d^* and select the domain with the lowest perplexity.

7.4.2 Information retrieval approach

The second method for text classification is based on the concept of information retrieval. Using the method described in [**?**], which is based on Equation **??**, we can calculate the similarity $\mathcal{S}_d(y, x)$ between a test document x and the development set y of a domain as

$$\mathcal{S}_d(y, x) = \sum_{f \in W_x \cap W_y} \frac{1}{(|W_x^f| + 1)|W_x|}, \tag{7.5}$$

where W_y is the set of words for the development set, W_x is the set of words in the test document, $|W_x|$ is the vocabulary size of the test document, and $|W_x^f|$ is the number of documents in the test corpus containing the word f. We select the domain with the highest score $\mathcal{S}_d(y, x)$ for each test document x.

Chapter 8

Results

In this chapter we introduce evaluation criteria and datasets used in this work as well as in the baseline translation systems. Experiments related to the approaches that are discussed in the main part in Chapter **??**, **??**, **??** and **??** will be described in this chapter to serve as a context for the discussion.

In comparison to the results we concentrate on the evaluation criteria described in Section **??** as a measure of translation performance, although current automatic translation evaluations do not perfectly match human evaluation results. This is disregarded in the basic comparison here.

We focus on Chinese-to-English translation systems, therefore experiments in this thesis only cover this single language pair. However, the Chinese word segmentation method in Chapter **??** is theoretically applicable to translate Chinese to any other language, and methods in Chapter **??**, **??** and **??** are generalized to translations between any language pair.

8.1 Evaluation criteria

If a translation is generated, a quantitative evaluation of the validity of this translation is necessary. Ideally, error counting by a human being would be suitable to score any output. However, human evaluation is expensive and time consuming, different evaluator might come to different conclusions and their decision is not always repeatable. Therefore, automatic evaluations which are based on comparing human generated references to a translation hypothesis are introduced in machine translation. Some of the current evaluation criteria are based on the edit distance [**?**], which can be categorized into two classes: error rates including the word error rate (WER), the position-independent word error rate (PER) [**?**] and the translation edit error rate (TER) [**?**]; other criteria are based on accuracy measures such as the BLEU score [**?**] and the NIST score [**?**].

Usually, a single reference cannot capture all formulations of a translation, therefore the above mentioned criteria are computed with respect to multiple references, where the average or minimum errors as well as precisions are counted with regard to these references. The translation results are measured case sensitive, while translation systems

are built in lower case, and the hypotheses are converted into true case using the SRI toolkit [?].

8.2 Task and corpus statistics

In this work experiments are performed on two types of datasets: a small data track, where the training corpus is rather clean and efficient to test the translation algorithms on sparse data; a large data track, where a better translation performance is expected. Here, the bilingual training corpus contains hundreds of million words, and the monolingual English data can obtain trillions of words.

We take the [?] (International Workshop on Spoken Language Translation) task for the small data track. The IWSLT organization holds an annual evaluation campaign that is carried out using a multilingual speech corpus on a small data track. It contains tourism-related sentences similar to those usually found in phrase books for tourists going abroad.

For the large data track we experiment on the [?] 2006, 2008 as well as on the [?] 2005, 2006, 2008 task. The GALE (Global Autonomous Language Exploitation) program is one of the currently well-known machine translation projects based on a very large amount of training data. In the annual evaluations the source language, either Arabic or Chinese, is translated into English. Input data will be in the form of either audio or text with the output always being text. The MT evaluation series started in 2001 as part of the [?] Translingual Information Detection, Extraction and Summarization (TIDES) program driven and coordinated by [?]. They provide an important contribution to the direction of research efforts and the calibration of technical capabilities in MT. The Open MT evaluations are intended to be of interest to all researchers working on the general problem of automatic translation between human languages.

The baseline systems are the official submission systems by RWTH-Aachen university in the evaluations of [?], [?] 2006, 2008 and [?] 2005, 2006, 2008. The best or well comparable results worldwide are also presented for each evaluation. In the following context we will present the corpus statistics of these tasks. The training corpus (Train) is used to train the word alignment and segmentation models. The feature weights of different translation models in the decoding are optimized on the development corpus (Dev) using the downhill simplex [?] algorithm with respect to the BLEU [?] score. The resulting systems are evaluated on the evaluation (Eval) corpora. In all large tracks, numbers, dates and hours are categorized and only the categorizations are contained in the statistics.

- IWSLT 2007

  The experiments of the Chinese word and phrase segmentation are performed on the [?] task. The *Basic Travel Expression Corpus* (BTEC) [?] is a multilingual speech corpus which contains tourism-related sentences similar to those found in phrase books. The corpus was provided in the course of the International Workshop on Spoken Language Translation [?].

Table 8.1: Corpus statistics of the IWSLT 2007 task.

		Chinese						English
		Chars	ICT-CLAS	LDC	Uni-gram	Learned-IU	Semi-CWS	
Train	Sentences[K]	42.9						
	R.W.[K]	519.9	380.3	385.4	393.8	343.7	396.8	420.4
	Vocabulary[K]	2.8	11.8	9.4	8.8	13.3	6.2	9.9
	Singletons	364	4637	2841	2629	4755	727	3937
Dev2	Sentences	500						
	R.W.[K]	4.8	3.6	3607	3.7	3.3	3.7	3.9
	Vocabulary	823	950	1021	987	1078	1004	834
	OOVs (R.W.)	7	75	52	49	17	16	216
	OOVs (Voc.)	6	73	50	47	15	14	44
Dev3	Sentences	506						
	R.W.[K]	5.2	3.8	3.8	3.9	3.6	4.0	4.0
	Vocabulary	837	936	996	969	1081	980	831
	OOVs (R.W.)	242	72	51	51	18	19	194
	OOVs (Voc.)	20	69	48	48	16	15	45
Eval	Sentences	489						
	R.W.[K]	4.3	3.2	3.3	3.3	3.0	3.4	3.8
	Vocabulary	762	885	944	915	1008	904	819
	OOVs (R.W.)	5	60	37	33	9	13	205
	OOVs (Voc.)	5	59	36	32	9	12	33

After the tokenization and automatic sentence segmentation, the training corpus nearly contains 43K bilingual sentences for each language as shown in Table **??**. We calculated the number of words and the vocabulary size as well as the number of singletons of the corpus.

As shown in Table **??** we used three test sets from the [**?**] translation evaluations: Dev2, Dev3 and Eval. Each of them contains 16 references. For convenience, we only list the statistics of the first reference translation after the tokenization. Dev2 is selected as the development corpus, Dev3 and Eval are taken as evaluation corpora. We show the statistics using different Chinese word segmentations: the standard CWS methods such as character-based translation (Chars), ICTCLAS [**?**], LDC [**?**] and Unigram, as well as the proposed method which is the learned segmentation with alignment combination IU [**?**] and semi-supervised CWS described in Section **??** and Section **??** respectively. Running words (R.W.), out of vocabulary in running words i.e. OOVs (R.W.) and out of vocabulary in vocabulary i.e. OOVs (Voc.) are listed, too.

- GALE 2008

  The experiments of Chinese word segmentation contributing to the system combination were performed on the [**?**] 2008 task and used in the final submission system.

Table 8.2: Corpus statistics of the GALE 2008 task.

		Chinese	English
Train	Sentences[M]	19.5	
	Running words[M]	242.9	264.3
	Vocabulary[K]	295.0	545.1
	Singletons[K]	122.5	247.7
Dev	Sentences	480	
	Running words[K]	13.8	17.2
	Vocabulary[K]	3.3	3.3
	OOVs (running words)	2	71
	OOVs (in vocabulary)	2	50
Test	Sentences	485	
	Running words[K]	13.7	17.1
	Vocabulary[K]	3.3	3.3
	OOVs (running words)	4	85
	OOVs (in vocabulary)	3	53

The training corpora for [?] are a collection of individual corpora collected from different sources and provided by the Linguistic Data Consortium [?] and [?]. The domains of most sub-corpora are news articles. Some sub-corpora contain documents from other domains such as transcriptions of broadcast conversation, web text and newsgroups.

The corpus statistics of the bilingual training data and the test sets are shown in Table **??**. The preprocessing step includes tokenization and the categorization of numbers and dates. Long sentences are segmented into short sentences using the sentence segmentation method that we introduced in Chapter **??** to reduce the training time. After the preprocessing and segmentation, the parallel training data contains more than 19.5 million sentences and more than 240 million words in each language.

- GALE 2006

  Earlier experiments of the domain dependent model combination for domain adaptation in Chapter **??** have been carried out on the [?] Chinese-English tasks of 2006. We discuss two domains of the test data: newswire and newsgroup.

  The language models were trained on the English part of the bilingual training corpus and on the monolingual data from the LDC GigaWord corpus. The total amount of the language model training data is around 1.5 billion running words.

  For domain dependent optimization of log-linear scaling factors in Chapter **??** we use the [?] 2002 evaluation set as the newswire and the [?] dry run development corpus as the newsgroup development set. The evaluation set is the [?] evaluation data. We aim to optimize the baseline system with 'in domain data' so that we have two domain specific SMT systems. One is trained discriminatively to translate newswire documents and the other is trained for newsgroup documents.

Table 8.3: Corpus statistics of task GALE 2006.

			Chinese	English
Train		Sentences[M]	20.3	
		Running words[M]	249	269
		Vocabulary[K]	251	430
		Singletons[K]	109	160
Dev	newswire	Sentences	878	
		Running words[K]	24.1	27.9
		Vocabulary[K]	4.1	3.9
		OOVs (running words)	3	100
		OOVs (vocabulary)	3	65
	newsgroup	Sentences	2 203	2 115
		Running words[K]	41.1	46.8
		Vocabulary[K]	5.7	5.4
		OOVs (running words)	11	113
		OOVs (vocabulary)	4	83
Eval	newwire	Sentences	460	364
		Running words[K]	10.0	10.3
		Vocabulary[K]	2.6	3.1
		OOVs (running words)	11	1 279
		OOVs (vocabulary)	9	875
	newsgroup	Sentences	441	415
		Running words[K]	9.6	10.5
		Vocabulary[K]	2.6	3.0
		OOVs (running words)	11	1 378
		OOVs (vocabulary)	8	989

Because of the large amount of training data and the categorization, the out-of-vocabulary words (OOVs) on all Chinese test sets are low. The statistics of the English references are measured without preprocessing.

- NIST 2008

  The experiments of the semi-supervised Chinese word segmentation is also performed on a large data track [**?**] 2008. The development corpus is a part of the evaluation data provided by [**?**] in 2006 (MT-06), and the evaluation corpus is the evaluation data in 2008 (MT-08). The corpus statistics are shown in Table **??**.

- NIST 2006

  The experiments of mixture language model adaptation are performed on the [**?**] 2006 task, where the proposed method was used in the final system to generate the RWTH Aachen University official submission results. Originally, there are about 9.5 million parallel sentences for the word alignment training. After the binary sentence splitting there are 19.5 million sentence pairs. More words are included if

Table 8.4: Corpus statistics of task NIST 2008.

		Chinese	English
Train	Sentences[M]	8.2	
	Running words[M]	192.1	205.2
	Vocabulary[K]	178.6	103.3
	Singletons[K]	64.0	725.5
Dev (MT-06 nist)	Sentences	1664	
	Running words[K]	40.5	46.2
	Vocabulary[K]	6.0	5.6
	OOVs (running words)	57	151
	OOVs (in vocabulary)	36	94
Eval (MT-08)	Sentences	1357	
	Running words[K]	34.4	42.3
	Vocabulary[K]	6.1	5.6
	OOVs (running words)	28	143
	OOVs (in vocabulary)	13	108

Table 8.5: Corpus statistics of task NIST 2006.

		Chinese	English
Train	Sentences[M]	9.5	
Train with sentence segmentation	Sentences[M]	19.5	
	Running words[M]	225.1	243.9
	Added running words	8.0%	8.2%
	Vocabulary[K]	236	400
Dev (MT-02)	Sentences	878	
	Running words[K]	24.1	27.9
	Vocabulary[K]	4.1	3.9
	OOVs (running words)	8	112
	OOVs (in vocabulary)	4	69
Test (MT-05)	Sentences	1 082	
	Running words[K]	32.1	34.4
	Vocabulary[K]	5.2	4.8
	OOVs (running words)	8	185
	OOVs (in vocabulary)	5	93
Eval (MT-06)	Sentences	3940	
	Running words[K]	87.2	103
	Vocabulary[K]	9.1	8.4
	OOVs (running words)	86	474
	OOVs (in vocabulary)	60	296

long sentences are not truncated. The evaluation data in 2002 (MT-02) is taken as a development corpus, and the evaluation data in 2005 (MT-05) and the evaluation data in 2006 (MT-06) are taken as two evaluation corpora. The corpus statistics on task [**?**] 2006 is shown in Table **??**.

- NIST 2005

  We will present the experiments on the sentence segmentation described in Chapter **??** on the translation task of [**?**] 2005.

  The training corpora used in the [**?**] tasks are a set of individual corpora provided by the Linguistic Data Consortium [**?**]; the domain is news articles. The translation experiments are carried out on the NIST 2002 evaluation set (MT-02).

  As shown in Table **??** the bilingual sentences are segmented into shorter segment pairs for an efficient training (Sentence segmented) and to include long sentences that are filtered out before (Sentence segmented all). We calculated the number of sentences and running words in the original and segmented corpora.

  There are 8.6 million sentence pairs in the original corpus, and the average sentence length is about 25. After the bilingual sentence segmentation described in Chapter **??** we have 17.9 million sentence pairs, and the average sentence length is around 12. The training time is reduced from one week to one and a half days. Due to a limitation of GIZA++ [**?**], sentences consisting of more than one hundred words are filtered out in the original corpus. Segmentation of long sentences circumvents this restriction and allows us to include more data. If we include the sentences that are too long to be used without segmentation, we obtain 19.5 million sentence pairs and thus we are able to add 8% Chinese and 8.2% English running words to the training data.

- FBIS

  For the sentence alignment task we need a document aligned corpus instead of a sentence aligned one. Therefore, we only take the Foreign Broadcast Information Service (FBIS) corpus, one of the sub-corpora from the [**?**] and [**?**] evaluation, to perform the translation. This corpus is document aligned and therefore we employ it for the experiments of the sentence alignment task.

  Table **??** shows the statistics of the FBIS corpus, which contains over 50 bilingual documents. Only document alignments are provided for this corpus. After applying the paragraph alignment described in Section **??** we have nearly 81 thousand paragraphs, 8.6 million Chinese and 10.1 million English running words. One of the advantages of the sentence segmentation is that we do not loose words during the extraction of bilingual sentences. However, we produce sentence pairs with very different lengths. Using Champollion we loose 10.8% of the Chinese and 3.1% of the English words.

There are four main topics in this thesis: Chinese word segmentation, phrase pair segmentation, sentence segmentation and document segmentation. In the following sections, we will present experiments for the methods proposed in this work in comparison

Table 8.6: Corpus statistics of task NIST 2005.

		Chinese	English
Train	Sentences[M]	8.6	
	Running words[M]	210	226
	Average Sentence Length	24.4	26.3
	Vocabulary[K]	224.3	359.6
	Singletons[K]	98.8	156.5
Train with sentence segmentation	Sentences[M]	17.9	
	Running words[M]	210	226
	Average sentence length	11.7	12.6
	Vocabulary[K]	221.5	353.1
	Singletons[K]	97.1	153.0
Train with sentence segmentation + additional data	Sentences[M]	19.5	
	Running words[M]	230.3	248.2
	Added running words	8.0%	8.2%
Eval (MT-02)	Sentences	878	3512
	Running words[K]	24.1	105.5
	Vocabulary[K]	4.1	6.8
	OOVs (Running words)	8	658
	OOVs (in vocabulary)	4	382

Table 8.7: Corpus statistics of task FBIS.

		Segmentation		Champollion	
		Chinese	English	Chinese	English
Train	Sentences[K]	739.9		177.8	
	Running words[M]	8.6	10.1	7.7	9.8
	Average sentence length	11.6	13.7	43.1	55.1
	Vocabulary[K]	34.9	56.6	34.4	55.8
	Singletons[K]	4.8	19.3	4.6	19.0
Eval (MT-02)	Sentences	878	3513	878	3513
	Running words[K]	24.1	105.5	24.1	105.5
	Vocabulary[K]	4.1	6.8	4.1	6.8
	OOVs (Running words)	109	2257	119	2309
	OOVs (in vocabulary)	59	882	66	891

to the worldwide and RWTH final submissions on the same tasks. The mapping between the methods and the tasks they were applied to is summarized in Table **??**. All algorithms of this work are implemented by myself except for the mixture language modeling using the SRI tool [**?**]. The algorithms are based on my publications during my Ph.D. study. For Chinese word segmentation we introduced a learned segmentation from alignment [**?**], semi-supervised Chinese word segmentation in training [**?**] and integrated Chinese word segmentation in search [**?**]. For phrase pair segmentation we introduced the mixture

Table 8.8: Algorithms and translation tasks for experiments.

Algorithm	Description in Chapter	Translation tasks
Semi-supervised CWS	Section ??	[?] 2008, [?] 2008, [?]
Integrated CWS	Section ??	[?]
Phrase segmentation	Chapter ??	[?]
Sentence segmentation	Chapter ??	[?] 2005, FBIS
Domain adaptation	Chapter ??	[?] 2006, [?] 2006

model framework and MER training as solutions to combine different features [?], the applied feature derivation based on IBM model 1 and HMM are previously described in [?] and [?] respectively. We introduced the sentence segmentation method with normalized IBM model 1 and its efficient calculation [?] as well as the log-linear model [?] to consider various features. For the document segmentation, the mixture language model was previously applied in automatic speech recognition. We firstly used this tool to generate language models for machine translation [?].

8.3 Chinese word segmentation

The experiments for the learned Chinese word segmentation in Section **??** are performed on the IWSLT 2007 task, see Table **??**, and the experiments for the semi-supervised Chinese word segmentation in Section **??** are performed on the IWSLT 2007 task, NIST 2008 task as well as GALE 2008 task, see Table **??** and Table **??**. The integrated Chinese word segmentation method described in Section **??** is applied to the IWSLT 2007 and NIST 2008 task. These approaches allow us to translate directly based on Chinese characters, because the Chinese word boundary detection is integrated into the translation processes. The word segmentation model is learned automatically from the bilingual training corpus during the training of the word alignment.

8.3.1 Statistics of the word length in the dictionary

The central idea of the learned and semi-supervised CWS methods is to automatically generate the lexicon using bilingual information so that the segmentation is task- and domain- oriented. As there is no unique definition of a 'correct' lexicon, we will compare the statistics on the word lengths of the learned lexicon and the word lengths of the lexicon generated by the semi-supervised CWS using Gibbs sampling (GS lexicon) to that in the manual lexicon provided by LDC [**?**].

Table 8.9: Statistics of word lengths in the vocabulary of the LDC lexicon, learned lexicon with alignment combination IU [**?**] and lexicon generated by semi-supervised CWS using Gibbs sampling (GS).

Word length	LDC lexicon		Learned lexicon IU		GS lexicon	
	Count	[%]	Count	[%]	Count	[%]
1	2 334	18.6	2 582	16.5	1 941	29.3
2	8 149	65.1	6 926	44.1	3 599	54.3
3	1 188	9.5	3 670	23.4	508	7.7
4	759	6.1	1 507	9.6	141	2.1
5	70	0.6	490	3.1	24	3.6
6	20	0.2	267	1.7	9	1.4
7	6	0.0	118	0.8	3	0.5
> 7	11	0.0	130	0.8	1	0.0
total	12 527	100	15 690	100	6226	100

Table **??** shows the statistics of the word lengths in the three lexicons. We calculate the number of word entries, distinguishing between the different lengths from 1 to 7 and the lengths larger than 7. For example, there are 2 334 words consisting of a single character in the LDC lexicon, 2 582 words in the learned lexicon and 1 941 words in the GS lexicon. These single character words represent 18.6% of the total number of entries in the LDC lexicon, 16.5% in the learned lexicon and 29.3% in the GS lexicon.

From Table ?? we see that in the manual LDC lexicon more than 60% of the words consist of two characters and only about 15% of the words consist of three or four characters. Longer words with more than four characters are used rarely. Evidently, there are too many words with more than two characters in the learned dictionary. In the GS lexicon, the length distribution is similar to that in the LDC lexicon. There are about 15% word entries containing more than two characters. Figure ?? visualizes the statistics in Table ??. The horizontal axis shows the word lengths and the vertical axis shows the percentage of the word entries in the lexicon with a given length.

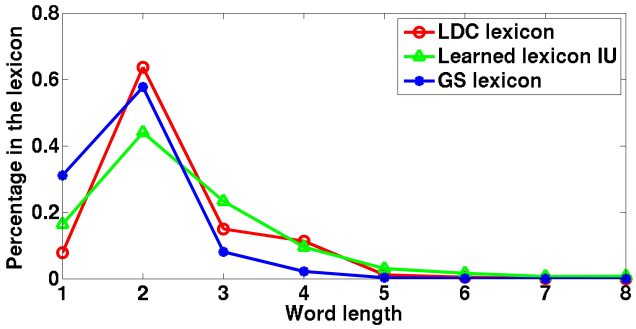


Figure 8.1: Number of words given a length in the LDC lexicon, the learned lexicon with alignment combination IU and the GS lexicon using Gibbs sampling.

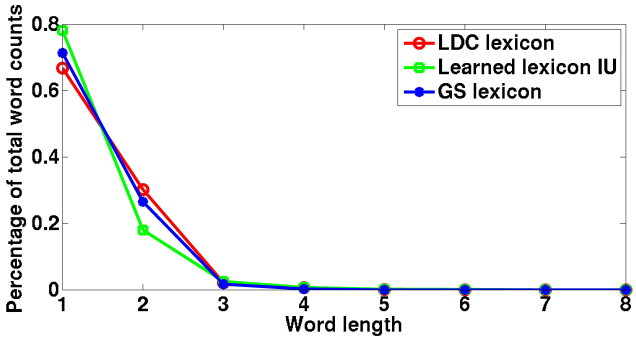


Figure 8.2: Total number of words given a length in the LDC lexicon, the learned lexicon with alignment combination IU and GS lexicon using Gibbs sampling.

Each word entry in the lexicon has a certain frequency. If we take these frequencies into account, we obtain the total number of words given a length in the lexicon. The percentages are shown in Figure ??. For example, if the frequency of one two-character word entry is ten, we add ten to the two-character words count instead of one. Figure ?? shows that the word distribution in the manual LDC lexicon is closer to that in the GS lexicon than to that in the learned lexicon.

8.3.2 Translation results

We show the translation results of the CWS methods in Table ??. This includes translation on characters i.e. each character is taken as a word, LDC [?], ICTCLAS [?]

Table 8.10: Translation results for various Chinese word segmentation methods studied in this work on the IWSLT 2007 task. The system using ICTCLAS is an improved RWTH system in Table **??** built after the final submission. The experiments marked by an asterisk are based on the software downloaded from [**?**] and [**?**].

Test	Method	BLEU[%]	TER[%]
Dev2	Unigram	53.9	38.6
	Semi-supervised CWS	55.1	37.5
	Integrated CWS	54.1	37.7
Dev3	Unigram	59.0	33.4
	Semi-supervised CWS	59.6	32.7
	Integrated CWS	58.4	32.5
Eval	Characters	38.0	47.0
	LDC*	37.9	46.2
	ICTCLAS*	38.8	46.0
	Learned-IU	38.6	46.6
	Unigram	40.2	46.6
	semi-supervised CWS	40.2	45.9
	Integrated CWS	39.3	45.3

Table 8.11: Comparison of this work with the first ten ranked official submission results at the IWSLT machine translation evaluation in 2007.

Organization	BLEU[%]
Institute for Infocomm Research, Singapore	40.8
Chinese Academy of Science, Beijing	37.5
Carnegie Mellon University, Pittsburgh	37.4
RWTH Aachen University	37.1
Institute of Automation at Chinese Academy of Sciences, Beijing	36.5
MIT Lincoln Lab and Air Force Research Lab, USA	36.3
Fondazione B. Kessler, Italy	34.7
Hong Kong University of Science and Technology	34.3
University of Maryland	32.1
ATR, Japan	31.3
This work (Semi-supervised CWS)	40.2

and the unigram method described in Section **??**, as well as the learned segmentation with IU alignment combination and semi-supervised method with an initialization of unigram segmentation using the manual LDC lexicon. The experiments marked by an asterisk are based on the softwares downloaded from [**?**] and [**?**]. Other experiments are performed with tools implemented by myself. The evaluations are performed under automatic criteria BLEU and TER scores with multiple references described in Section **??**.

The semi-supervised CWS is evaluated using the full model with both monolingual and bilingual information according to Equation **??**. The model weights λ in Equation **??** are optimized using the Powell [**?**] algorithm with respect to the BLEU score. We obtained $\lambda_1 = 1.4$, $\lambda_2 = 1$ and $\lambda_3 = 0.8$ as optimal values and $T = 4$ as the optimal number of iterations of the re-alignment with GIZA++ in Table **??**.

From Table **??** we see that the Chinese word segmentation using semi-supervised CWS leads to the best translation performance on IWSLT 2007 according to the BLEU and TER score. In the BLEU score the translation performance is 1.4% higher than that of the ICTCLAS and in the TER score it is 0.7% lower than that of the unigram method. We also show the translation results using integrated Chinese word segmentation.

In Table **??** we list the first ten ranked official submission results from the different participants, which include the Institute for Infocomm Research in Singapore (I^2R), the Institute of Computing Technology at Chinese Academy of Sciences in China (ICT), Carnegie Mellon University in Pittsburgh (CMU), RWTH Aachen University in Germany (RWTH), the Institute of Automation at Chinese Academy of Sciences in China (CASIA), the MIT Lincoln Laboratory and the Air Force Research Laboratory Wright-Patterson AFB (MIT-LL+AFRL), the Fondazione B. Kessler in Italy (FBK), the Hong Kong University of Science and Technology (HKUST), the University of Maryland (UMD) and the ATR Spoken Language Communication Research Laboratories in Japan (ATR).

Table 8.12: Comparison of this work with results reported by Google and the RWTH's final submission on NIST 2008[%].

System	Dev 08		Test 08	
	BLEU	TER	BLEU	TER
One of Google's systems [**?**]	-	-	28.5	-
Final submission by RWTH (LDC CWS)	33.4	60.6	26.2	65.6
This work (semi-supervised CWS)	34.2	60.1	26.4	65.4
This work (integrated-CWS)	33.5	61.0	26.2	66.6

We also show translation experiments performed on NIST 2008. Because of the intellectual property issue, we cannot repeat the experiments on the evaluation set of 2008 and compare it with any official result by participants outside RWTH. We choose one of Google's systems described in [**?**] as a worldwide baseline. As an internal baseline, LDC Chinese word segmentation is applied in the official final submission system of RWTH Aachen University. The system using the semi-supervised CWS outperforms the RWTH baseline system by 0.2% in the BLEU score and 0.2% in the TER score absolutely.

In the semi-supervised CWS we take the unigram method to initialize the Gibbs sampling and to segment the test corpus using a combined lexicon as illustrated in Figure **??**. The probability of each word entry from the manual lexicon is linearly interpolated with that in the Gibbs sampling generated lexicon. The weight for the manual lexicon is 0.6, and the weight for the Gibbs sampling generated lexicon is 0.4. The integrated Chinese word segmentation does not perform better than the baseline in the large data track, because we are currently not able to put corpora using different Chinese word segmentations into

Table 8.13: Final translation submission results of leaving-one-system-out experiment in system combination by RWTH on GALE 2008 for newswire data[%].

Newswire	Dev (Test 08)		Eval (Dev 08)	
	BLEU	TER	BLEU	TER
Best single system	31.0	61.6	32.1	61.8
System combination	34.9	57.8	35.7	57.5
Leaving one out: without pbt-LDC	35.2	57.7	35.6	57.6
Leaving one out: without pbt-semi-supervised-CWS	34.6	58.0	35.3	57.8

word alignment training due to limited computational resources and a large amount of training data.

To measure the diversity of translation systems generated based on different Chinese word segmentations, we performed a leaving-one-system-out experiment. In Table **??** we show the translation performance by remaining a single system from the system combination and re-optimizing the weights on the Test 08 data (Dev). The Dev 08 data was used as a blind test set (Eval). The experiment was performed on the newswire documents.

The result on the blind test by the best single system at RWTH official submission is evaluated with a BLEU score of 32.1%. Several systems are included in the system combination: standard phrase-based systems, syntax-based systems and systems with different Chinese word segmentations. After the system combination, the BLEU score is improved to 34.9%. By leaving out the phrase-based system with LDC segmentation, the BLEU score decreases 0.1%, while by leaving out the phrase-based system with semi-supervised CWS, the BLEU score decreases 0.4%, which indicates that the system with semi-supervised CWS contributes 0.4% in the BLEU score on Dev 08 in the RWTH system combination. The semi-supervised CWS is applied to the system combination for the final submission of RWTH Aachen University.

8.3.3 Analysis of translation outputs

We present some examples of translation output to show that the segmentation has an effect on the translation quality in Table **??**. Three examples are selected from the experiments on the evaluation corpus of IWSLT 2007. For each of them we show the Chinese source sentence segmented using the baseline unigram, learned segmentation and semi-supervised segmentation method, as well as their corresponding translation and the human reference translation.

In the first example, both semi-supervised CWS and baseline methods lead to correct segmentations, while the learned segmentation results in an error, because '晚些' should be separated into two words. '晚' means 'late' and '些' means 'a little'.

In the second example, the translation results in the learned and semi-supervised CWS segmentation are closer to the reference translation. As the single character words '金' and '额' occur more often than their combination in the training corpus, it is easier to

Table 8.14: Examples of segmentation and translation outputs with baseline, learned segmentation and semi-supervised CWS method.

a)	Baseline	我 会 晚 些 到 但 请 不要 取消 我 的 预定 。
		i will arrive late , but please do not cancel my reservation .
	Learned-IU	我 会 晚些 到 但 请 不要 取消 我 的 预定 。
		i will call later , but please do not cancel my reservation .
	Semi-supervised	我 会 晚 些 到 但 请 不要 取消 我 的 预定 。
	CWS	i will arrive late , but please do not cancel my reservation .
	REF	I'll arrive late , but keep my reservation , please .
b)	Baseline	请 告诉 我 总 金额。
		please show me the in .
	Learned-IU	请 告诉 我 总 金 额 。
		please show me the total price .
	Semi-supervised	请 告诉 我 总 金 额 。
	CWS	please show me the total price .
	REF	can you tell me the total amount ?
c)	Baseline	请 给 我 可口可乐 。
		please give me .
	Learned-IU	请给 我 可口 可乐 。
		please give me a good coke .
	Semi-supervised	请 给 我 可 口 可乐 。
	CWS	please give me a coke .
	REF	coke , please .

recognize the single character word in the evaluation text.

In the third example, the segmentation with both learned method and baseline method made mistakes. For the learned method '请给' should be two words, where '请' means 'please' and '给' means 'give'. Though the baseline segmentation is reasonable for a human evaluation, the translation result is still erroneous because the sequence of characters '可口可乐' never appears in the training corpus, but '可乐' can be found many times. As both of them mean 'coke', we only need the word '可乐' to obtain the correct translation.

We manually compared the translation output on the evaluation set using semi-supervised CWS with the baseline method. 196 sentences are different out of 489 lines whereby 64 sentences from semi-supervised CWS are better, 33 sentences are worse, and the remaining sentences have a similar translation quality.

8.3.4 Conclusions

We have successfully developed novel Chinese word segmentation methods for statistical machine translation. In the training process, Chinese word boundaries are learned jointly with the word alignments. Both monolingual and bilingual information are employed to derive a segmentation suitable for MT. New Chinese words and their distributions are generated automatically. At translation time, multiple segmentation alternatives

instead of the single-best segmentation are considered, and the segmentation decision is taken during the search for the best translation. The semi-supervised CWS in training outperforms the standard Chinese word segmentation approach in terms of translation quality.

8.4 Phrase pair segmentation

In this section we will discuss the translation experiments of the phrase pair segmentation methods described in Chapter **??**.

8.4.1 Translation results

Table 8.15: Translation results for various phrase pair segmentation methods studied in this work on the IWSLT 2007 task. 'Baseline' is presented in Table **??**; 'Baseline + HMM features' combines two additional features derived from HMM posterior probabilities as described in Section **??**; 'Baseline + IBM model 1 phrases' includes phrase pairs extracted using IBM model 1 posterior probabilities as in Section **??** into the standard phrase table.

Test	Method	BLEU	TER
Dev (Dev2)	Baseline	53.9	38.6
	Baseline + HMM features	54.3	37.7
	Baseline + IBM model 1 phrases	55.5	37.1
Eval (Dev3)	Baseline	59.0	33.4
	Baseline + HMM features	59.2	33.1
	Baseline + IBM model 1 phrases	59.6	32.6

Table **??** gives us an overview of the translation results with the baseline phrase extraction method described in Section **??** as well as with the mixture model in Section **??**. We use the standard phrase extraction method to obtain phrase pairs in 'Baseline'. In 'Baseline + HMM feature' phrase pairs are evaluated with two additive features, which are computed using HMM posterior probabilities in normal and inverse direction as described in Section **??**. The factors of the HMM features are optimized together with the scaling factors of the log-linear model in the decoding using minimum error rate, see [**?**].

For the development and evaluation data, a set of evaluation scores is presented in Table **??**. On the development corpus, using the mixture method, the TER is decreased by nearly 1% absolute, and the BLEU score is improved, too. We take Dev3 as evaluation data, which is considered more similar in content to the development corpus Dev2. On the evaluation data the PER is reduced by 0.4% absolutely. The BLEU score is increased, and the WER and PER are decreased, too. We see that adding the HMM posterior probabilities leads to better translation results compared to the baseline method on all test sets under all evaluation criteria.

Table **??** also shows experimental results on combining the standard phrase extraction method with the method using the IBM model 1 posterior probabilities described in Section **??**. Probabilities of phrase pairs that are extracted using the standard method are combined linearly with that of phrase pairs extracted using IBM model 1 with a weight of 0.4. In the row of 'Baseline + IBM model 1 phrases', we can see that on all

test sets with respect to different criteria, the improvements of mixture model over the conventional approach are stable.

The experimental results of phrase segmentation on a large data track is not shown in this work, because the computational requirements to extract phrase pairs using the proposed methods are too large so that it is not feasible to perform at the moment.

8.4.2 Analysis on translation outputs

Table 8.16: Translation outputs with baseline and mixture methods.

a)	Chinese	要交钱吗？
	Baseline	it will cost ?
	Mixture	do i have to pay for this ?
	REF	do i have to pay ?
b)	Chinese	是樱花盛开的季节了。
	Baseline	is cherry blossoms .
	Mixture	it is cherry blossoms .
	REF	it's cherry blossom season .
c)	Chinese	我和约翰在一起工作。
	Baseline	i am john .
	Mixture	i am with john together .
	REF	i work with john together .
d)	Chinese	怎么拼写？
	Baseline	the door will not close ?
	Mixture	how can i spell ?
	REF	how do you spell it ?
e)	Chinese	能再给我一瓶啤酒吗？
	Baseline	can i have a beer please ?
	Mixture	could you give me a bottle of beer ?
	REF	could i get another bottle of beer ?

We manually compared the translation output of the system using the mixture model with the baseline system. On the development data the translations of 148 sentences are different out of 500 lines, where 40 sentences from mixture model are translated better, 17 sentences are translated worse, and the rests have similar translation quality. Table **??** shows five examples from the development corpus. We list source and reference sentences as well as the translations produced by the baseline and mixture methods. Though some translations are not optimal, the mixture model generates better translation results than the baseline method does in general.

8.4.3 Conclusions

The generic phrase training algorithm follows an information retrieval perspective but aims to improve both precision and recall with the trainable log-linear model. A clear advantage of the proposed approach over the widely used phrase extraction method based on Viterbi word alignments is the trainability. Under the general framework one can put together as many features as possible under the log-linear model to evaluate the quality of a phrase pair. The phrase table extraction procedure is trainable and can be optimized jointly with the translation engine to directly maximize the end-to-end translation performance. Another advantage is flexibility, which is provided partially by the threshold. We use feature functions to decide the order and the threshold to locate the boundary guided by a development set. We investigated which features are important and valuable in ranking candidate phrase pairs. We propose several feature functions derived from posterior distribution. The standard phrase extraction method is a special case where a single binary feature function defined from word alignments is used. The experimental results on IWSLT 2007 have demonstrated a consistent improvement over the widely used extraction method based on the word alignment matrix.

8.5 Sentence segmentation

Table 8.17: Sentence segmentation vs. sentence alignment.

	Sentence segmentation	Sentence alignment
Method	Sentence segmentation as in Section ??	Hybrid approach as in Section ?? (Champollion+sentence segmentation)
Goal	Shorten sentences Reduce training time and memory Enlarge training data in use	Extract bilingual sentences Better translation performance
Task	NIST 2005	FBIS

Sentence segmentation and sentence alignment are closely related to each other. Bilingual sentence segmentation is concerned with to chop a long sentence pair into two sub-sentence pairs, which represents two problems: 1. how to locate the optimal split point. 2. how to assign the two sub-sentences in the target language to the sub-sentences in the source language. Sentence alignment only covers the second problem i.e. how to correctly assign sentences in the target language to sentences in the source language. However in this work we prove that the sentence segmentation method can also be applied to a sentence alignment task, where we treat the paragraph or document pair as a 'very long' bilingual sentence and chunk them until the sub-sentences are short enough.

Table **??** shows the methodologies, goals and tasks we applied for sentence segmentation and alignment respectively. We use the segmentation model described in Section **??** for sentence segmentation and the hybrid approach discussed in Section **??** to align sentences. The goal of the former is to reduce the computational requirement and include long sentences that are truncated before. The purpose of the latter is to extract bilingual sentences from document aligned corpora so that more training data can be included to train word alignment. We are going to apply the experiments of sentence segmentation on the NIST 2005 task and the experiments of sentence alignment on the FBIS task:

1. On the [**?**] 2005 task, we will show improvements using the sentence segmentation method as described in Section **??**. Besides a better translation performance, the usage of the sentence segmentation method also has other advantages:

   - Enlargement of data in use
     By splitting long sentences during preprocessing, fewer words are filtered out, as shown in Table **??**. Thus, we are able to use more data in the training.
   - Speedup of the training process
     In experiments the computational requirements of the word alignment training with GIZA++ can be significantly reduced after segmenting long sentence pairs.

2. We will compare the different sentence alignment methods described in Section **??** on the FBIS corpus with respect to translation performance. We do not apply the extraction methods on the whole NIST corpora, because most corpora provided by LDC [**?**] are already sentence aligned.

8.5.1 Segmentation parameters

The parameters of the segmentation model are optimized on some development data with respect to translation results measured on a system trained on the Treebank corpus, which is a subset of the NIST corpus. The length normalization factor β is set to $\beta = 0.9$, which configures the weight of the length normalization in Equation **??**. The minimum and maximum sentence lengths restrict the lengths of the sub-sentences within a range. We took a minimum length of 1 and a maximum length of 25.

We did not optimize the log-linear model scaling factors for the segmentation in Equation **??** but used the following fixed values:$\lambda_1 = \lambda_2 = 0.5$ for the IBM model 1 in both directions; $\lambda_3 = 10^8$, if the anchor words model is used; $\lambda_4 = 30$, if the IBM model 4 Viterbi word alignment is used.

8.5.2 Translation results

Table 8.18: Translation results for various sentence segmentation methods studied in this work on MT-02 test data for NIST 2005.

	BLEU[%]	TER[%]
Baseline	33.5	59.9
+ sentence segmentation	33.5	59.9
+ segmentation + concatenation	33.6	59.8
+ segmentation + concatenation + IBM model 4	33.6	59.8
+ segmentation + added data	33.9	59.6

Table 8.19: Comparison of this work with results reported by other research groups on MT-02 test data for NIST 2005.

	BLEU[%]
Google	36.0
IRST	27.9
Edinburgh	27.2
This work	33.9

For the segmentation of long sentences into short segments, we performed the experiments on the NIST task. Here, the NIST 2002 test set with 878 sentences is the evaluation corpus. We present the results of different experiments for sentence segmentation as shown in Table **??**, which is compared with worldwide results as shown in Table **??**:

1. **Baseline**: We filter out those sentences that contain more than one hundred words on either side of the language from the original training corpus to train the system.

2. **+ sentence segmentation**: We use exactly the same data that is actually used in the 'baseline' experiment, but apply the proposed splitting algorithm. Thus, the original training corpus is filtered and then split.

3. **+ segmentation + concatenation**: As described in Section **??** the word alignments trained on the split data are concatenated. The original source, target sentences and the concatenated alignments are used in phrase extraction.

4. **+ segmentation + concatenation + IBM model 4**: As described in Section **??** the Viterbi word alignment of model-4 is applied in the segmentation model. This method is only used if the data has been trained before and the word alignments are available. For example, to optimize the word alignment training parameters, we need to retrain the word alignments in shorter time.

5. **+ segmentation + added data**: Here, we *first* split the training corpus and *then* apply the filtering. This enables us to use more data because sentences that would have been removed in the 'baseline' experiment are now included. Note that still some sentences are filtered out because the source and target length differ too much.

Table 8.20: Time of training process and memory requirements for NIST 2005.

Method	Time (day)	Memory (GB)
Baseline	5.8	2.4
Sentence segmentation	1.4	1.2

Both in the baseline and the segmentation systems we obtain 4.7 million bilingual phrases during the translation. The method of alignment concatenation increases the number of the extracted bilingual phrase pairs from 4.7 million to 4.9 million; the BLEU score is improved by 0.1%. The IBM model 4 Viterbi word alignment does not help to improve the translation quality in the experiment. As shown in Table **??** the training of the baseline system requires more than five days. After the sentence segmentation it only requires less than one and a half days. Moreover, the segmentation allows the inclusion of long sentences that are filtered out in the baseline system. Including the additive data the translation performance is enhanced by 0.3% in BLEU. Because of the long translation period the translation parameters are only optimized on the baseline system with respect to the BLEU score. We could expect further improvement if the parameters were also optimized on the segmentation system.

One of the major objectives here is to introduce another approach to parallel sentence extraction: segmenting the bilingual texts recursively. We use the paragraph-aligned corpus as a starting point as illustrated in Figure **??**. Table **??** presents the translation results on the training corpora generated by the different sentence alignment methods described in Section **??**. We observe that the segmentation method is comparable to Champollion, and the segmentation with anchors outperforms the one without anchors. By combining the methods of Champollion and the segmentation with anchors, the BLEU score is improved by 0.4% absolutely.

Table 8.21: Translation results of the sentence alignment task using the FBIS training corpus on the MT-02 data set. The experiment marked by an asterisk is based on the software downloaded from [?].

	TER[%]	BLEU[%]
Champollion*	60.7	29.8
Segmentation preferring anchor words	60.4	30.1
Hybrid	60.5	30.3

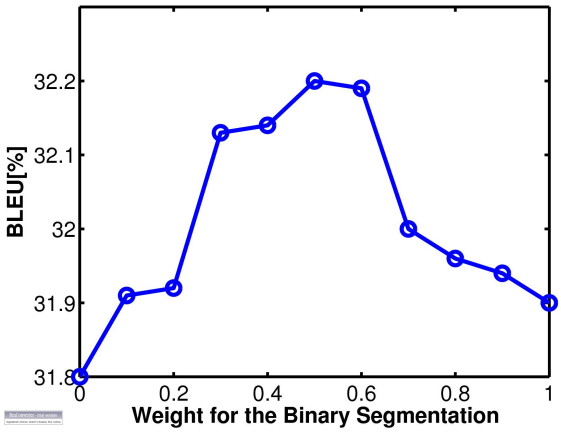


Figure 8.3: Translation performance as a function of the weight for the sentence segmentation (binary segmentation) α; the weight for Champollion is $1 - \alpha$.

We optimized the weight for the segmentation method in the hybrid approach; the sum of the weights for both methods is one. As shown in Figure **??** using one of the methods alone does not produce the best result. The maximum BLEU score is attained if both methods are combined with equal weights.

8.5.3 Conclusions

We have developed a new method to segment long bilingual sentences into several short parts using a modified IBM model 1 for an efficient word alignment training. The experiments on NIST 2005 task show both the reduction of the training time and the feasibility of long sentence pairs in training. We also successfully applied the sentence segmentation method as well as a hybrid method to extract bilingual sentence pairs from the document aligned texts. The experiments on the FBIS data for sentence alignment task show an improvement of 0.4% of the BLEU score compared to the score obtained using a state-of-the-art sentence aligner. In addition to the encouraging results obtained, further improvements could be achieved in the following ways:

1. Sentence parts without translation:
   In some bilingual sentences, one or more parts of a sentence in the source or target

language may have no translation at all. These parts should be marked or removed during the splitting process.

2. Alignment of non-consecutive sub-sentences:
   In the sentence segmentation we do not allow the alignment of non-consecutive segments. For example, the source sentence could be divided into three parts and the target sentence into two parts. The first and the third part of the source sentence might be translated as the first part into the target sentence, and the second part in the source sentence could be translated as the second part in the target sentence. Such a case is not yet discussed here.

3. By extracting bilingual paragraphs from the documents we lost running words using Champollion. Applying the segmentation approach to paragraph alignment might avoid the loss of this data.

8.6 Document segmentation

The experiments for domain adaptation have been carried out on the GALE 2006 and NIST 2008 Chinese-English task. The experiments for domain dependent model combination in Section **??** were performed on GALE 2006. There are two domains in this task: newswire and newsgroup. The domain of the test data is not given, so we applied the domain adaptation methods described in Section **??**. The experiments for domain dependent language modeling in Section **??** are performed on NIST 2008. The first use of the mixture language model in machine translation [**?**] enhances the translation performance significantly.

8.6.1 Classification results

The primary goal of addressing the domain issue in machine translation is to improve the translation quality. As the domain adaptation is implemented as document classification, the classification accuracy can be indicative of the translation performance. In this section we are going to present the results of the classification methods described in Section **??**.

We separate the evaluation set into two different domains: newswire and newsgroup. There are 55 documents in the evaluation set including 36 newswire and 19 newsgroup articles. We calculate the classification error rate by dividing the number of incorrectly classified documents by the number of all test documents.

1. Language model approach

   This method was presented in Section **??**. We build a six-gram language model using SRI toolkit [**?**] from the Dev newswire corpus and from the Dev group corpus respectively. Because of the limited resources in each domain we also produce a trigram general language model LM_g as in Equation **??** to cover some unknown words from the evaluation data. The vocabulary is constrained to the union of the vocabularies of the Dev newswire and Dev group. The general language model was trained on the Chinese side of the bilingual corpora. The lowest error rate is 25.5%, if the value of α (see Equation **??**) is set between 0.5 and 0.7. Experiments show that as long as a very high value is not given to the general language model weight, the results of the combination of the in-domain and the general language model are stable.

2. Information retrieval approach

   Using the information retrieval approach described in Section **??**, we have a classification error rate of 34.5%, where none of the newswire articles is classified incorrectly, and 19 of the newsgroup articles are classified incorrectly as newswire.

As the language model approach outperforms the information retrieval approach in the test document domain classification results, we simply perform the translation with the classification results obtained using the language model approach. The weight of the general language model is fixed to 0.5.

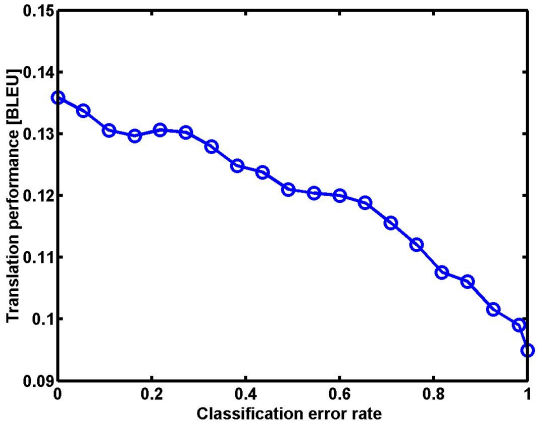


Figure 8.4: The translation performance in the BLEU score related to the documents classification error rate for newsgroup text with domain specific optimization for GALE 2006.

We plot the BLEU scores of the newsgroup text translations and the classification error rates measured for newsgroup text. The incorrectly classified documents are randomly selected. As shown in Figure **??**, a roughly proportional relationship exists between the document classification accuracy and the translation performance.

8.6.2 Translation results

On the GALE 2006 task we distinguish the systems with different settings of the scaling factors of the log-linear model in the decoder. In Table **??** the scaling factors of the baseline system are optimized on the newswire development corpus. If all newsgroup documents are translated with the optimized feature weights on the newsgroup development corpus, we receive oracle best translation results. Here, we show the oracle best results optimized with respect to BLEU and TER. Using the language model based on the document classification method with $\alpha = 0.5$, the BLEU score rises from 9% to 11%, which is an improvement of 18% relatively, while the TER score reduces, too. The oracle best shows the BLEU score can still reach to 13.6%, if all documents are classified correctly. The results in Table **??** are compared with worldwide results as illustrated in Table **??**.

Table 8.22: Translation results for various domain specific optimization methods on newsgroup text for GALE 2006.

Method		WER[%]	PER[%]	BLEU[%]	NIST	TER[%]
Baseline		73.3	62.0	9.5	3.0	70.0
Oracle best	Opt-bleu-newsgroup	77.6	59.1	13.6	4.6	71.3
	Opt-ter-newsgroup	75.6	61.4	9.4	3.6	68.4
Opt-bleu $\alpha = 0.5$		73.7	60.8	11.0	3.8	69.5

Table 8.23: Comparison of this work with results reported by other research groups for GALE 2006. The NIGHTINGALE group includes the RWTH team.

Method	TER[%]
AGILE	65.5
NIGHTINGALE	69.9
ROSETTA	71.7

Table 8.24: Translation results with various mixture language models for NIST 2006.

	TER[%]	BLEU[%]	LM PPL[%]	Memory requirement[GB]	
4-gram standard LM	65.0	26.5	103.0	< 16	< 4
6-gram mixture LM (RWTH without rescoring,postprocessing)	62.8	28.7	87.2	< 16	< 8

The experiments of domain dependent language models were applied at the NIST 2006 machine translation evaluation, see Table **??**. In Table **??**, we presented the first four ranked official results submitted by the Information Sciences Institute (ISI), Google, Language Weaver and RWTH Aachen University. In the RWTH submission system we applied a 6-gram mixture language model, the BLEU score measured on MT-06 test data is 30.2%. Without rescoring and postprocessing, the BLEU score is 28.7%. At the NIST 2006 evalaution, the maximal available computer memory for us was 16 GB. Due to this limitation we could only train a 4-gram language model using the standard approach. The mixture language model can reduce the memory requirement of the language model training significantly. By applying the mixture model we are able to train a 6-gram language model, which leads to a much lower perplexity, from 103.0 to 87.2, and a significant improvement in the translation quality i.e. about two percent in the BLEU score.

Table **??** shows the case insensitive results on the MT-05 test data. In the fourth column we see that the perplexity of the language model decreases rapidly by increasing the order of its history. The mixture language model allows to realize building larger language models with the same available computational resource. This leads to an absolute improvement of about two percent in the translation performance measured by the BLEU score, while the maximal memory in decoding increases from 4 GB to 8 GB.

8.6.3 Conclusions

We have discussed the domain issue in statistical machine translation and proposed the document segmentation method to build domain specific machine translation systems. We have used a combination of feature weights of the phrase-based log-linear translation

Table 8.25: Comparison of this work with results reported by other research groups for NIST 2006.

	BLEU[%]
ISI	33.9
Google	33.2
Language Weaver	32.8
This work (6-gram mixture LM)	30.2

Table 8.26: Translation experiments with various genre language models on MT-05 test set for NIST 2006, case insensitive

	TER[%]	BLEU[%]	LM PPL[%]	Memory requirement[GB]	
				LM training	Translation
4-gram standard LM	58.2	33.4	103.0	< 16	< 4
5-gram mixture LM	57.0	34.4	94.7	< 16	< 6
6-gram mixture LM	56.1	35.4	87.2	< 16	< 8
+Rescore (final system)	55.6	36.7	87.2	< 16	< 8

model to distinguish between multiple domains. The training of a domain specific system is a tuning process where the translation performance is to be maximized on a small amount of the domain specific development set which is a small amount of data. Moreover, the domain dependent language modeling enhances the translation performance significantly.

Domain classification during the translation of the test documents is implemented as monolingual text classification. We compared an approach based on language model to an approach based on information retrieval. We observed that the former one achieved a lower document classification error rate.

The results on the GALE 2006 Chinese-English translation task have shown that the domain adaption in the translation process achieved significant improvements over the domain independent translation, even in the case of a rather high document classification error rate in the domain adaptation. The mixture language model greatly improved the translation performance on the NIST 2006 task.

The future work is to better exploit document classification algorithms and to perform adaptation driven by the evaluation data.

Chapter 9

Scientific contributions

The main contributions of this work include the following issues (all presented algorithms are self-developed and implemented if not indicated otherwise):

- **Chinese word segmentation**

  Words in Chinese texts are usually segmented using an off-the-shelf method, and a standard translation model is applied given the fixed segmentation. Well-known methods are for instance LDC [**?**], ICTCLAS [**?**] and conditional random field [**?**]. Nonetheless, those methods are not necessarily optimal for translations due to the following two reasons: 1. The segmentations may be erroneous because the context varies. 2. The best segmentation for a given character sequence also depends on its translation. We propose word segmentation methods that are designed for the machine translation application, and segmentation ambiguities are considered during the search for the best translation.

  First, we train word segmentation and alignment models simultaneously, where both monolingual and bilingual information are applied to derive a segmentation suitable for machine translation. New words are introduced using Bayesian learning. On the [**?**] task, the best submission result is evaluated with a BLEU score of 40.8%. The semi-supervised Chinese word segmentation method improves the RWTH baseline system with ICTCLAS segmentation from 38.8% to 40.2% in the BLEU score. On the [**?**] 2008 task, one of Google's systems performed 28.5% on the evaluation data in 2008. The proposed method enhances the BLEU score from 26.2% to 26.4% over the official baseline submission system by RWTH Aachen University.

  Secondly, a lattice representing all segmentation alternatives is taken as an input of decoding instead of the single-best segmentation used in common approaches. Multiple word segmentations are considered, and the segmentation decisions are performed during the generation of translations. In this way, we are able to train and translate on sequences of Chinese characters. The experiments on the [**?**] task shows an improvement of translation performance from 53.9% to 54.1% in the BLEU score on the development set and from 38.8% to 39.3% on the evaluation set.

- **Sentence segmentation and improved sentence alignment**

The performance of data-driven machine translation systems heavily relies on the quantity and quality of the training data. Exploiting and efficiently employing this large amount of data is hence a crucial problem. The standard systems take parallel sentences to train translation models without chunking long sentence pairs. This results in a long training time. We apply sentence segmentation to reduce the computational requirement in the training. Previously, [?] searches for the segmentation boundaries using a dynamic programming algorithm. This technique only allows monotone sentence alignments, and manually defined anchor words are needed as possible segmentation boundaries.

We introduce a sentence segmentation method based on the ITG [?] bilingual parsing concept, which chops a long sentence pair into shorter segment pairs allowing segment reordering. Experiments on the [?] 2005 task show sentence segmentation is capable to speed up the word alignment training significantly. The translation quality of the system using the sentence segmentation is same as that of the baseline system. The BLEU score is 33.5% measured on the evaluation data in 2002 (The BLEU score of the Google system was 36% at that time). However, after applying sentence segmentation, the word alignment training time reduces from 5.8 days to 1.4 days and the memory requirement decreases from 2.4 GB to 1.2 GB. In addition, the proposed sentence segmentation method improves the performance of a state-of-the-art sentence aligner Champollion [?] for the translation application from 29.8% to 30.3% in the BLEU score on the evaluation data in 2002 of the FBIS task.

- **Phrase pair segmentation**

  State-of-the-art statistical machine translation uses phrases as translation units to incorporate a context into translation models, as described by [?], [?] and [?]. However, it is still far from a fair and comprehensive evaluation on phrase entries. The frequency model heavily depends on the Viterbi word alignment input, and a mistake in the Viterbi alignment results in errors in the phrase pair extraction. The information from the alignment models is not fully explored and is discarded after the output of the Viterbi word alignment.

  We present a generic phrase training model to minimize the translation error rates. The model is parameterized with feature functions and can be optimized jointly with the translation engine to maximize the end-to-end system performance directly. Multiple data-driven feature functions are able to capture the quality and confidence of phrases and phrase pairs. The methods in [?] and [?] are re-implemented as feature functions. Experimental results demonstrate consistent improvements over the widely used method that is based on the word alignment matrix only. On the [?] task, the BLEU score is enhanced from 59.0% to 59.6% on the test corpus.

- **Document segmentation**

  While translation performance has been advanced substantially in general, translation style and the domain issue leave much room for further improvements. Previous investigations have been applied to speech recognition such as [?]. We [?] address domain issues for statistical machine translation and propose the combination of feature weights and language model adaptation to distinguish

between multiple domains. The proposed approach requires much less parallel data than what is typically used to build a domain independent system, which makes it easy and efficient to capture as many domains as required. We realize a hierarchical clustering algorithm to segment a large corpus into different domains and use the SRI toolkit [**?**] to combine models from various domains. The results on the [**?**] 2006 task show improvements with the proposed domain dependent translation over domain independent translation. The TER score decreases from 70.0% to 68.4%, which is to be compared with 69.9% for NIGHTINGALE group and 71.7% for ROSETTA group and 65.5% for AGILE group. On the [**?**] 2008 task, the mixture language model enhances the translation performance significantly, from 26.5% to 28.7% in the BLEU score, using the same amount of memory.

Chapter 10

Conclusion

In the previous chapters we discussed several methods to model sequence segmentation aiming at a reduction of the translation error rate in statistical machine translation. We discussed the impacts of Chinese word segmentation, phrase pair induction as well as bilingual sentence and document segmentation in statistical machine translation. Various algorithms are presented to solve these problems. Both on translation performance and efficiency we achieved significant improvements in the following areas:

We showed that it is possible to learn Chinese word boundaries so that the translation performance of Chinese-to-English MT systems is enhanced. We presented a Bayesian generative model for parallel Chinese-English sentences which treats word segmentation as a hidden variable and incorporates both monolingual and bilingual information for word alignment and segmentation training. Starting with an initial word segmentation, the semi-supervised Chinese word segmentation learns both new Chinese words and distributions for these words using Bayesian learning. Not only in a small, but also in a large data environment, the proposed method outperformed the standard Chinese word segmentation approach in terms of the final Chinese-to-English translation quality.

Then we introduced and implemented a novel phrase extraction framework, which induces the phrase pairs using the word alignment model directly instead of from the single best word alignment. Multiple features and resources are combined for phrase pairs induction. We also unified the features in the phrase extraction and evaluation processes so that the phrase scoring is consistent and expanded to both processes. This turns the phrase pair inventory problem into a problem of how to assign the probabilities to the phrase pairs in a bilingual sentence more accurately and more fairly. Better phrase pairs are obtained by incorporating HMM and IBM model 1 derived information. The experiments verify that the mixture phrase pair model improves the state of the art machine translation system stably in a small data track.

Moreover, we successfully applied the binary segmentation method to split long sentence pairs for an efficient model training and extraction of bilingual sentence pairs from the document aligned texts. The experiments show the reduction of the word alignment training time and memory requirements as well as the enhancement of translation performance with an improved sentence alignment method.

Finally, we discussed the domain issue and proposed an efficient method to build

domain dependent machine translation systems by document segmentation. We used a combination of feature weights of the phrase-based log-linear translation model as well as the domain dependent mixture language modeling for system building. Domain specific training data is obtained by an unsupervised clustering method. A significant improvement has been achieved using domain adaptation.

Mathematical symbols

$\hat{\cdot}$	a selection from all candidates
$\alpha_1, \alpha_2, \alpha_3$	parameters in the linear combination for new word generation
β	sentence length normalization factor in modified IBM model 1
$\gamma_\theta(\cdot)$	the word posterior probability given model parameters θ
θ	model parameters
κ	a parameter of the spelling model in the semi-supervised CWS
λ	model scaling factor i.e. feature weight
σ	threshold to terminate an optimization process of feature weights in a mixture language model
τ	threshold to filter phrase pairs
ω_1, ω_2	a variable or a sequence of variables
φ_f	maximum allowed source phrase length
φ_e	maximum allowed target phrase length
$\psi(\mathcal{D}_1^R, \mathcal{D'}_1^{R'})$	function to tell if the clustering fullfills the termination condition
$\Upsilon(\cdot, \cdot)$	sentence segmentation algorithm
$a_1^J = a_1, ..., a_j, ..., a_J$	word alignment (mapping)
$b_1^I = b_1, ..., b_i, ..., b_I$	inverse word alignment (mapping)
c_k^+	the segmentation where a boundary exists after c_k
c_k^-	the segmentation where no boundary exists after c_k
$cba_{k,e\_}^+$	segmentation c_k^+ along with the product space of relevant alignments in both directions $b_{k,e\_}^+$ and a_k^+
cba_k^-	segmentation c_k^- along with the product space of relevant alignments in both directions b_k^- and a_k^-
$\|c\|$	character vocabulary size
$d_s = (c_1^K, e_1^I)$	a bilingual sentence pair i.e. an observation in the semi-supervised CWS
dh_{sk}^-	all observations and hidden variables which are not involved by c_k-th position in s-th sentence pair
$\tilde{e} = e_{i_1}^{i_2} = e_{i_1}, ..., e_{i_2}$	target phrase
$\tilde{f} = f_{j_1}^{j_2} = f_{j_1}, ..., f_{j_2}$	source phrase
$e\_ = 1, ..., \|e\|$	index of possible alignments after splitting a Chinese word
$\|e\|$	number of English words aligned to a Chinese word
h	hidden variables
$h(\cdot)$	feature function of a log-linear or mixture model
i	a word position in a sentence of the target language
j	a word position in a sentence (document) of the source language
$k_1^J = k_1, ..., k_j, ..., k_J$	positions of Chinese word boundaries of sentence c_1^K
$m \in \{1, ..., M\}$	feature function index
n	language model order
o	alignment orientation in sentence segmentation
$q \in \{1, ..., Q\}$	document index
$r \in \{1, ..., R\}$	domain i.e. cluster index

$s = 1, ..., S$	index of parallel sentences in the training data
$t = 1, .., T$	iteration index in a training algorithm
u_q	q-th document
v	number of characters in a document i.e. character vocabulary size
x	test document
y	development document
$A \subseteq J \times I$	word alignment (general)
$A^{i_1,i_2}_{j_1,j_2}$	the set of word alignment that aligns source phrase $f_{j_2}^{j_1}$ to target phrase $e_{j_1}^{i_1}$
$\mathcal{A}$	user defined anchor word list for sentence segmentation
$B \subseteq I \times J$	word alignment inverse (general)
$C = c_1^K = c_1, ..., c_k, ..., c_K$	source language sentence in characters
$D = (d_1, ..d_s, .., d_S)$	a bilingual training corpus
$\mathcal{D}_r$	r-th domain i.e. cluster
$\|\mathcal{D}_r\|$	the number of documents in cluster r
$E = e_1^I = e_1, ..., e_i, ..., e_I$	target language sentence in words
$F = f_1^J = f_1, ..., f_j, ..., f_J$	source language sentence in words
$H(\cdot)$	function to estimate bilingual entropy
I	length of a target language sentence in words
I_{max}/I_{min}	maximum/minimum target sentence length in the sentence segmentation
J	length of a source language sentence (or document) in words
J_{max}/J_{min}	maximum/minimum source sentence length in sentence segmentation
K	length of a source language sentence in characters
L	word length
N	total number of Chinese words in the training corpus
$N(f)$	frequency of Chinese word f in the previous context
$N(f, e)$	coocurrence frequency of f and e in the previous context
$N_a(\cdot)$	number of Viterbi word alignments
$O(\cdot)$	complexity
$P(\cdot)$	model-based probability distribution
$P_0(f)$	prior for Chinese word f
$P_d(f\|\cdot)$	domain specific language model probability
$P_d^*(f\|\cdot)$	interpolated language model probability
$P_{ef}(\cdot)$	translation probability in the inverse direction for semi-supervised CWS
$P_{fe}(\cdot)$	translation probability in normal direction for semi-supervised CWS
$P_g(f\|\cdot)$	general language model probability
$P_m(\cdot)$	monolingual word probability for semi-supervised CWS
$P_G(f)$	probability of word f according to distribution G
$P_{G_e}(f\|e)$	probability of f given e according to distribution G_e.
$Pr(\cdot)$	general probability distribution with no specific assumptions
$\mathcal{S}(\mathcal{D}_{r_1}, \mathcal{D}_{r_2})$	similarity measure between cluster r_1 and r_2
$\mathcal{S}_d(x, y_r)$	similarity between a test set and the development set of domain r

W_r	the set of unique content words in cluster r
$\|W_r\|$	the number of words in W_r
Z	a normalization factor

Acronyms

BLEU	bilingual evaluation understudy
BTEC	basic travel expression corpus
C-to-E	Chinese-to-English
CWS	Chinese word segmentation
Dev	development corpus
DARPA	Defense Advanced Research Projects Agency
E-to-C	English-to-Chinese
Eval	evaluation corpus
EM	expectation maximization
FBIS	Foreign Broadcast Information Service
GALE	Global Autonomous Language Exploitation
GS	Gibbs sampling
HMM	hidden Markov model
IBM	International Business Machines Corporation
ICT	Institute of Computing Technology
ITG	inversion transduction grammar
IU	intersection and union
IWSLT	International Workshop on Spoken Language Translation
LDC	Linguistic Data Consortium
LM	language model
MCMC	Markov chain Monte Carlo
MER	minimum error rate
MT	machine translation
NIST	National Institute of Standards and Technology
NLP	natural language processing
OOV	out-of-vocabulary
PER	position-independent word error rate
PPL	perplexity
R.W.	running words
RWTH	Rheinisch-West-fälischen Technischen Hochschule
SMT	statistical machine translation
TER	translation edit rate
TIDES	Translingual Information Detection, Extraction, and Summarization
Voc.	vocabulary
WER	word error rate

List of Tables

4.1 An example for the definition of Chinese words and word segmentations. . 16

4.2 All possible word segmentations for c_1^K are illustrated in Table **??**. The number of all possible segmentations for c_1^K is 2^{K-1}. The best word segmentation is the sentence with ID. 2. 18

4.3 Manually generated Chinese word lexicon. 18

4.4 An example of translation hypotheses of a Chinese sentence with different Chinese word segmentations. 19

4.5 Observations and hidden variables of the generative model for Chinese word segmentation. 25

4.6 General algorithm of Gibbs sampling for CWS. 31

4.7 Complete algorithm of Gibbs sampler for CWS including alignment models. The observations are $D = (d_1, ..d_s, .., d_S)$, where d_s=(c_1^K, e_1^I) indicates a bilingual sentence pair. Hidden variables F_t and A_t indicate the word segmentation and word alignment of the corpus in the t-th iteration respectively. 35

4.8 An example of simulating the process of the integrated Chinese word segmentation. 39

4.9 A word list generated from the vocabulary of the Chinese training corpus. 41

5.1 MER training for mixture feature weights in phrase pair segmentation. . . 50

6.1 Recursive binary sentence segmentation procedure. 54

6.2 Efficient computation of IBM model 1. 57

7.1 Stepwise optimal hierarchical clustering using bottom-up method (agglomerative hierarchical clustering) where $\mathcal{D}_r$ indicates the r-th domain ($r \in 1, .., R$), and u_q indicates q-th document ($q \in 1, .., Q$). 62

7.2 EM algorithm of feature weights λ optimization in a unigram mixture language model. $\lambda_r \in \{1, .., R\}$: the weight of the r-domain; $j \in \{1, .., J\}$: a word position in the development corpus; σ: a user-defined threshold. . . 66

8.1 Corpus statistics of the IWSLT 2007 task. 70

8.2 Corpus statistics of the GALE 2008 task. 71

8.3 Corpus statistics of task GALE 2006. 72

8.4 Corpus statistics of task NIST 2008. 73

8.5 Corpus statistics of task NIST 2006. 73

8.6 Corpus statistics of task NIST 2005. 75

8.7 Corpus statistics of task FBIS. 75

8.8 Algorithms and translation tasks for experiments. 76

8.9 Statistics of word lengths in the vocabulary of the LDC lexicon, learned lexicon with alignment combination IU [**?**] and lexicon generated by semi-supervised CWS using Gibbs sampling (GS). 77

8.10 Translation results for various Chinese word segmentation methods studied in this work on the IWSLT 2007 task. The system using ICTCLAS is an improved RWTH system in Table **??** built after the final submission. The experiments marked by an asterisk are based on the software downloaded from [**?**] and [**?**]. 79

8.11 Comparison of this work with the first ten ranked official submission results at the IWSLT machine translation evaluation in 2007. 79

8.12 Comparison of this work with results reported by Google and the RWTH's final submission on NIST 2008[%]. 80

8.13 Final translation submission results of leaving-one-system-out experiment in system combination by RWTH on GALE 2008 for newswire data[%]. . 81

8.14 Examples of segmentation and translation outputs with baseline, learned segmentation and semi-supervised CWS method. 82

8.15 Translation results for various phrase pair segmentation methods studied in this work on the IWSLT 2007 task. 'Baseline' is presented in Table **??**; 'Baseline + HMM features' combines two additional features derived from HMM posterior probabilities as described in Section **??**; 'Baseline + IBM model 1 phrases' includes phrase pairs extracted using IBM model 1 posterior probabilities as in Section **??** into the standard phrase table. . . 84

8.16 Translation outputs with baseline and mixture methods. 85

8.17 Sentence segmentation vs. sentence alignment. 87

8.18 Translation results for various sentence segmentation methods studied in this work on MT-02 test data for NIST 2005. 88

8.19 Comparison of this work with results reported by other research groups on MT-02 test data for NIST 2005. 88

8.20 Time of training process and memory requirements for NIST 2005. 89

8.21 Translation results of the sentence alignment task using the FBIS training corpus on the MT-02 data set. The experiment marked by an asterisk is based on the software downloaded from [**?**]. 90

8.22 Translation results for various domain specific optimization methods on newsgroup text for GALE 2006. 93

8.23 Comparison of this work with results reported by other research groups for GALE 2006. The NIGHTINGALE group includes the RWTH team. 94

8.24 Translation results with various mixture language models for NIST 2006. . 94

8.25 Comparison of this work with results reported by other research groups for NIST 2006. 95

8.26 Translation experiments with various genre language models on MT-05 test set for NIST 2006, case insensitive 95

List of Figures

4.1 An example of an alignment matrix between Chinese characters and English words. A black box indicates a single-best (Viterbi) word alignment. 21

4.2 Workflow of semi-supervised CWS. 24

4.3 Transition from no boundary(−) to a boundary(+): The monolingual probability $P_m(cba_k^+|dh_{sk}{}^-)$ is estimated as $P_G(f')P_G(f'')$, and the translation probability in the E-to-C direction $P_{ef}(cba_k^+|dh_{sk}{}^-)$ is estimated as $\frac{1}{J+2}{}^{|e|}P(e_\prime|f')P(e_{\prime\prime}|f'')$ or $\frac{1}{J+2}{}^{|e|}P(e_{\prime\prime}|f')P(e_\prime|f'')$ or $\frac{1}{J+2}{}^{|e|}P(e_\prime|f')P(e_\prime|f'')$ or $\frac{1}{J+2}{}^{|e|}P(e_{\prime\prime}|f')P(e_{\prime\prime}|f'')$, here $|e|=2$ and $J=2$. 32

4.4 Transition from a no-boundary(−) to a boundary(+): The monolingual probability $P_m(cba_k^+|dh_{sk}{}^-)$ is estimated as $P_G(f')P_G(f'')$, and the translation probability in the C-to-E direction $P_{fe}(cba_k^+|dh_{sk}{}^-)$ is estimated as $\frac{1}{I+1}{}^2P_G(f'|e_*)P_G(f''|e_*)$, here $|e|=2$ and $I=2$. 32

4.5 Transition from a boundary(+) to no boundary(−): The monolingual probability $P_m(cba_k^-|dh_{sk}{}^+)$ is estimated as $P_G(f)$, and the translation probability in the E-to-C direction $P_{ef}(cba_k^-|dh_{sk}{}^+)$ is estimated as $\frac{1}{J}{}^{|e|}P_G(e'|f)P_G(e''|f)$, here $|e|=2$,$J=3$. 33

4.6 Transition from a boundary(+) to no boundary(−): The monolingual probability $P_m(cba_k^-|dh_{sk}{}^+)$ is estimated as $P_G(f)$, and the translation probability in the C-to-E direction $P_{fe}(cba_k^-|dh_{sk}{}^+)$ is estimated as $\frac{1}{I+1}P_G(f|e'_*)$ or $\frac{1}{I+1}P_G(f|e''_*)$, here $|e|=2$ and $I=3$. 33

4.7 An example of a word segmentation and alignment alternatives using Gibbs sampling. 35

4.8 Translation procedures of the integrated Chinese word segmentation vs. single-best word segmentation. 37

4.9 Single-best segmentation: the input sentence as a linear automaton 39

4.10 Segmentation lattice composed of a manual and a character-based segmentation 40

4.11 Segmentation transducer. 41

4.12 Segmentation lattice without weights including all word segmentation alternatives given a vocabulary. 42

4.13 Three segmentations composed of a character-based segmentation, a manual segmentation and an automatic segmentation weighted by the word length model if $\eta = 1$. 42

4.14 Segmentation lattice weighted by a language model considering all alternatives given a vocabulary. 43

5.1 An example of phrase pair segmentation using the standard approach: the left figure shows that a wrong word alignment results in a missing phrase pair; the right figure shows that this missing phrase pair can be generated using the correct word alignment. 44

5.2 Architecture of mixture phrase table vs. standard phrase table generation. 46

5.3 Phrase extraction using IBM model 1 based on the posterior probabilities of word alignments. The posterior probability of a phrase pair alignment between $f_{j_1}, .., f_{j_2}$ and $e_{i_1}, .., e_{i_2}$ is defined as the sum of the posterior probabilities of the word alignments in the shaded areas. 47

5.4 Phrase extraction using HMM model based on the posterior probabilities of word alignments. The posterior probability of a phrase pair alignment between $f_{j_1}, .., f_{j_2}$ and $e_{i_1}, .., e_{i_2}$ is defined as the sum of posterior probabilities of the word alignments in the shaded area after normalization. 48

6.1 Two types of sentence alignment in binary sentence segmentation. 53

6.2 Word alignment matrix of a sentence pair, where darker blocks indicate a lexicon probability. The shaded area indicates the alignments of the sub-sentence pairs after the first iteration of segmentation. 58

6.3 Result of the sentence segmentation example. 58

6.4 Sentence alignment using binary sentence segmentation method, dynamic programming algorithm (Champollion) and a hybrid approach. 59

8.1 Number of words given a length in the LDC lexicon, the learned lexicon with alignment combination IU and the GS lexicon using Gibbs sampling. . 78

8.2 Total number of words given a length in the LDC lexicon, the learned lexicon with alignment combination IU and GS lexicon using Gibbs sampling. 78

8.3 Translation performance as a function of the weight for the sentence segmentation (binary segmentation) α; the weight for Champollion is $1 - \alpha$. 90

8.4 The translation performance in the BLEU score related to the documents classification error rate for newsgroup text with domain specific optimization for GALE 2006. 93

Bibliography

[Aldous 85] D. Aldous: Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII-1983*, pp. 1–198, Springer, Berlin, Germany, 1985.

[Andrew 06] G. Andrew: A hybrid Markov/semi-Markov conditional random field for sequence segmentation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pp. 465–472, Sydney, Australia, July 2006.

[Bellegarda 00] J. Bellegarda: Exploiting latent semantic information in statistical language modeling. In *Proceedings of IEEE*, number 8, pp. 1279–1296, 2000.

[Besling & Meier 95] S. Besling, H. Meier: Language model speaker adaptation. In *Proceedings of European Conference on Speech Communication and Technology*, pp. 1755–1758, Madrid, Spain, 1995.

[Blunsom & Osborne 08] P. Blunsom, M. Osborne: Probabilistic inference for machine translation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pp. 215–223, Honolulu, HI, October 2008.

[Brown & Lai$^+$ 91] P.F. Brown, J.C. Lai, R.L. Mercer: Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 177–184, Berkeley, CA, June 1991.

[Chen & Seymore$^+$ 98] S.F. Chen, K. Seymore, R. Rosenfeld: Topic adaptation for language modeling using unnormalized exponential models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 681–684, Seattle, WA, May 1998.

[Chen & Zhou$^+$ 05] A. Chen, Y. Zhou, A. Zhang, G. Sun: Unigram language model for Chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pp. 138–141, Jeju Island, Korea, October 2005.

[Chen 93] S.F. Chen: Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31th Annual Meeting of the Association for Computational Linguistics*, pp. 9–16, Columbus, OH, June 1993.

[Church & Hanks 91] K. Church, P. Hanks: Word association norms, mutual information and lexicography. *Computational linguistics*, Vol. 16(1), pp. 22–29, 1991.

[DARPA 09] DARPA. Defense Advanced Research Projects Agency, 2009. http://www.darpa.mil.

[Daumé III & Marcu 04] H. Daumé III, D. Marcu: A phrase-based HMM approach to document/abstract alignment. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pp. 119–126, Barcelona, Spain, July 2004.

[Daumé III & Marcu 06] H. Daumé III, D. Marcu: Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, Vol. 26(1), pp. 101–126, May 2006.

[Deng & Kumar$^+$ 07] Y. Deng, S. Kumar, W. Byrne: Segmentation and alignment of parallel text for statistical machine translation. *Natural Language Engineering*, Vol. 13(3), pp. 235–260, 2007.

[Deng 05] Y. Deng. *Bitext Alignment for Statistical Machine Translation*. Ph.D. thesis, Johns Hopkins University, Baltimore, MD, 2005.

[Doddington 02] G. Doddington: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology Conference*, pp. 128–132, San Diego, CA, March 2002.

[Duda & Hart$^+$ 01] R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification*. John Wiley & Sons, Inc., New York, 2nd edition, 2001.

[Foster & Kuhn 07] G. Foster, R. Kuhn: Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 128–135, Prague, Czech Republic, June 2007.

[GALE] GALE. NIST Machine Translation Evaluation for GALE Home Page. http://www.nist.gov/speech/tests/gale.

[Gale & Church 93] W.A. Gale, K.W. Church: A program for aligning sentences in bilingual corpora. *Computational Linguistics*, Vol. 19(1), pp. 75–90, 1993.

[Galescu & Allen 00] L. Galescu, J. Allen: Hierarchical statistical language models: experiments on in-domain adaptation. In *Proceedings of the International Conference on Spoken Language Processing*, pp. 186–189, Beijing, China, October 2000.

[Gao & Li$^+$ 05] J. Gao, M. Li, A. Wu, C. Huang: Chinese word segmentation and named entity recognition: a pragmatic approach. *Computational Linguistics*, Vol. 31(4), pp. 531–574, 2005.

[Gao & Suzuki$^+$ 06] J. Gao, H. Suzuki, W. Yuan: An empirical study on language model adaptation. *ACM Transactions on Asian Language Information Processing*, Vol. 5(3), pp. 209–227, 2006.

[Geman & Geman 84] S. Geman, D. Geman: Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6(6), pp. 721–741, 1984.

[Goldwater & Griffiths$^+$ 09] S. Goldwater, T.L. Griffiths, M. Johnson: A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, Vol. 112(1), pp. 21–54, 2009.

[ICT] ICT. ICTCLAS Chinese word segmentation system home page.

[IWSLT 05] IWSLT. International Workshop on Spoken Language Translation home page, 2005. http://www.is.cs.cmu.edu/iwslt2005.

[IWSLT 07] IWSLT. International Workshop on Spoken Language Translation home page, 2007. http://www.is.cs.cmu.edu/iwslt2007.

[Iyer & Ostendorf 96] R. Iyer, M. Ostendorf: Modeling long distance dependence in language: topic mixtures vs. dynamic cache models. In *Proceedings of the Fourth International Conference on Spoken Language Processing*, pp. 236–239, Philadelphia, PA, October 1996.

[Kanthak & Ney 04] S. Kanthak, H. Ney: FSA: an efficient and flexible C++ toolkit for finite state automata using on-demand computation. pp. 510–517, Barcelona, Spain, July 2004.

[Khadivi 08] S. Khadivi. *Statistical Computer-Assisted Translation*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, July 2008.

[Kim & Zhang$^+$ 01] S. Kim, B. Zhang, Y.T. Kim: Learning-based intrasentence segmentation for efficient translation of long sentences. *Machine Translation*, Vol. 16(3), pp. 151–174, 2001.

[Kim 05] W. Kim. *Language model adaptation for automatic speech recognition and statistical machine translation*. Ph.D. thesis, Johns Hopkins University, Baltimore, MD, 2005.

[Koehn & Och$^+$ 03] P. Koehn, F.J. Och, D. Marcu: Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference*, pp. 127–133, Edmonton, Canada, May/June 2003.

[LDC 03] LDC. Linguistic Data Consortium Chinese resource home page, 2003. http://www.ldc.upenn.edu/Projects/Chinese.

[LDC 05a] LDC. Champollion Tool Kit 1.0 Home Page, 2005. http://sourceforge.net/projects/champollion.

[LDC 05b] LDC. Linguistic Data Consortium resource home page, 2005. http://www.ldc.upenn.edu/Projects/TIDES.

[Low & Ng$^+$ 05] J.K. Low, H. Ng, W. Guo: A maximum entropy approach to Chinese word segmentation. pp. 161–164, Jeju Island, Korea, October 2005.

[Luo & Roukos 96] X. Luo, S. Roukos: An iterative algorithm to build Chinese language models. In *Proceedings of the 34th annual meeting of the Associaton for Computational Linguistics*, pp. 139–143, Santa Cruz, CA, 1996.

[Ma 06] X. Ma: Champollion: A robust parallel text sentence aligner. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pp. 489–492, Genoa, Italy, May 2006.

[Mahajan & Beeferman$^+$ 99] M. Mahajan, D. Beeferman, X.D. Huang: Improved topic-dependent language modeling using information retrieval techniques. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 541–544, Phoenix, AZ, March 1999.

[Marcu & Wong 02] D. Marcu, W. Wong: A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pp. 133–139, Philadelphia, PA, July 2002.

[Moore 02] R.C. Moore: Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the Fifth Conference of the Association for Machine Translation in the Americas*, pp. 135–244, Tiburon, CA, October 2002.

[Munteanu & Marcu 05] D. Munteanu, D. Marcu: Improving machine translation performance by exploiting comparable corpora. In *Computational Linguistics*, Vol. 31(4), pp. 477–504, December 2005.

[Nelder & Mead 65] J. Nelder, R. Mead: A simplex method for function minimization. *Computer Journal*, Vol. 7(4), pp. 308–313, 1965.

[Nevado & Casacuberta$^+$ 03] F. Nevado, F. Casacuberta, E. Vidal: Parallel corpora segmentation by using anchor words. In *Proceedings of the European Association for Machine Translation and Association for Computational Linguistics European Chapter Workshop on MT and Other Language Technology Tools*, pp. 12–17, Budapest, Hungary, April 2003.

[NIST] NIST. NIST open machine translation evaluation home page. http://www.nist.gov/speech/tests/mt.

[Och & Ney 04] F.J. Och, H. Ney: The alignment template approach to statistical machine translation. *Computational Linguistics*, Vol. 30(4), pp. 135–244, December 2004.

[Och 99] F.J. Och: An efficient method for determining bilingual word classes. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 71–76, Bergen, Norway, June 1999.

[Och 00] F.J. Och. GIZA++: Training of statistical translation models, 2000. http://www-i6.informatik.rwth- aachen.de/~och/software/GIZA++.html.

[Och 02] F.J. Och. *Statistical machine translation: from single word models to alignment templates*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, October 2002.

[Och 03] F.J. Och: Minimum error rate training in statistical machine translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*, pp. 160–167, Sapporo, Japan, July 2003.

[Palmer 97] D.D. Palmer: A trainable rule-based algorithm for word Segmentation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 321–328, Madrid, Spain, 1997.

[Papineni & Roukos$^+$ 02] K.A. Papineni, S. Roukos, T. Ward, W.J. Zhu: Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, PA, July 2002.

[Press & Teukolsky$^+$ 02] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery: *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK, 2002.

[Seymore & Chen$^+$ 98] K. Seymore, S. Chen, R. Rosenfeld: Nonlinear interpolation of topic models for language model adaptation. In *Proceedings of the Fifth International Conference on Spoken Language Processing*, pp. 2503–2506, Sydney, Australia, 1998.

[Shannon 48] C.E. Shannon: A mathematical theory of communication. *The Bell System Technical Journal*, Vol. 27, pp. 379–423,623–656, July,October 1948.

[Simard & Langlais 03] M. Simard, P. Langlais: Statistical translation alignment with compositionality constraints. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Canada, May 2003.

[Snover & Dorr$^+$ 06] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul: A study of translation edit rate with targeted human annotation. In *Proceedings of the Seventh Conference of the Association for Machine Translation in the Americas*, pp. 223–231, Cambridge, MA, August 2006.

[Sproat & Shih 90] R.W. Sproat, C. Shih: A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, Vol. 4(4), pp. 336–351, April 1990.

[Stolcke 02] A. Stolcke: SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference On Spoken Language Processing*, pp. 901–904, Denver, CO, September 2002.

[Sun & Shen+ 98] M. Sun, D. Shen, B.K. Tsou: Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pp. 1265–1271, Montreal, Quebec,Canada, August 1998.

[Takezawa & Sumita+ 02] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, S. Yamamoto: Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pp. 147–152, Las Palmas, Spain, May 2002.

[Theodoridis & Kourtroumbas 06] S. Theodoridis, K. Kourtroumbas: *Pattern Recognition*. Elsevier, USA, 2006.

[Tillmann & Ney 00] C. Tillmann, H. Ney: Word re-ordering and DP-based search in statistical machine translation. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 850–856, Saarbrücken, Germany, July 2000.

[Tromble & Kumar+ 08] R.W. Tromble, S. Kumar, F. Och, W. Macherey: Lattice minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pp. 620–629, Hawaii, HI, 2008.

[Ueffing & Haffari+ 07] N. Ueffing, G. Haffari, A. Sarkar: Semi-supervised model adaptation for statistical machine translation. *Machine Translation*, Vol. 21(2), pp. 77–94, 2007.

[Venugopal & Vogel+ 03] A. Venugopal, S. Vogel, A. Waibel: Effective phrase translation extraction from alignment models. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pp. 319–326, Sapporo, Japan, 2003.

[Vogel & Hewavitharana+ 04] S. Vogel, S. Hewavitharana, M. Kolss, A. Waibel: The ISL statistical translation system for spoken language translation. In *Proceedings of the International Workshop on Spoken Language Translation 2004*, pp. 65–72, Kyoto, Japan, September 2004.

[Vogel 05] S. Vogel: PESA: phrase pair extraction as sentence splitting. In *Proceedings of the Tenth Machine Translation Summit*, pp. 251–258, Phuket, Thailand, September 2005.

[Wang & Liu 05] Z. Wang, T. Liu: Chinese unknown word identification based on local bigram model. *International journal of computer processing of oriental languages*, Vol. 18(3), pp. 185–196, 2005.

[Wu & Wang+ 08] H. Wu, H. Wang, C. Zong: Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 993–1000, Manchester, UK, August 2008.

[Wu 97] D. Wu: Stochastic grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, Vol. 23(3), pp. 377–403, September 1997.

[Xu & Deng+ 07] J. Xu, Y. Deng, Y. Gao, H. Ney: Domain dependent machine translation. In *Proceedings of the Eleventh Machine Translation Summit*, pp. 78–85, Copenhagen, Danmark, September 2007.

[Xu & Gao+ 08] J. Xu, J. Gao, K. Toutanova, H. Ney: Bayesian semi-supervised Chinese word segmentation for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 1017–1024, Manchester, UK, August 2008.

[Xu & Matusov$^+$ 05] J. Xu, E. Matusov, R. Zens, H. Ney: Integrated Chinese word segmentation in statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pp. 141–147, Pittsburgh, PA, October 2005.

[Xu & Zens$^+$ 04] J. Xu, R. Zens, H. Ney: Do we need Chinese word segmentation for statistical machine translation? In *Proceedings of the Third SIGHAN Workshop on Chinese Language Learning*, pp. 122–128, Barcelona, Spain, July 2004.

[Xu & Zens$^+$ 05] J. Xu, R. Zens, H. Ney: Sentence segmentation using IBM word alignment model 1. In *Proceedings of the Tenth Annual Conference of the European Association for Machine Translation*, pp. 280–287, Budapest, Hungary, May 2005.

[Xu & Zens$^+$ 06] J. Xu, R. Zens, H. Ney: Partitioning parallel documents using binary segmentation. In *Proceedings of the Joint Conference of Human Language Technology and the North American Chapter of the Association for Computational Linguistics Workshop on Statistical Machine Translation*, New York City, NY, June 2006.

[Zens & Ney 04] R. Zens, H. Ney: Improvements in phrase-based statistical machine translation. In *Proceedings of the Joint Conference of Human Language Technology and North American Chapter of the Association for Computational Linguistics*, pp. 257–264, Boston, MA, May 2004.

[Zens & Och$^+$ 02] R. Zens, F.J. Och, H. Ney: Phrase-based statistical machine translation. In *Proceedings of the 25th German Conference on Artificial Intelligence*, pp. 18–32, Aachen, Germany, September 2002. Springer Verlag.

[Zhang & Malik 03] H. Zhang, J. Malik: Learning a discriminative classifier using shape context distances. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 242–247, Madison, WI, June 2003.

[Zhang & Vogel$^+$ 03] Y. Zhang, S. Vogel, A. Waibel: Integrated phrase segmentation and alignment algorithm for statistical machine translation. In *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, pp. 567– 573, Beijing, China, October 2003.

[Zhao & Eck$^+$ 04] B. Zhao, M. Eck, S. Vogel: Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th International Conference on Computational Linguistics*, pp. 411–417, Geneva, Switzerland, 2004.

Curriculum Vitae

Contact Information

Chair of Computer Science VI
Professor Dr.-Ing. H. Ney
RWTH Aachen, Ahornstr. 55
D-52056 Aachen, Germany

Voice: +49 (241) 80-21617
Fax: +49 (241) 80-22219
E-mail: xujia@cs.rwth-aachen.de
http://www-i6.informatik.rwth-aachen.de/∼xujia

Education

- **RWTH Aachen University**, Aachen, Nordrhein-Westfalen Germany 08/2003 - present

  Ph.D. Candidate, Computer Science
- **Technical University of Berlin**, Berlin Germany 10/2000 - 04/2003

  Diploma, Computer Science
- **Nankai Middle School**, Tianjin China 07/1994 - 07/1997

  Higher School Certificate

Work Experience

Ph.D. Candidate, 08/2003 - present
RWTH Aachen University, Aachen
- Research topic: "Chinese - English Statistical Machine Translation"
- Supervisor: Prof. Hermann Ney

Diploma candidate, 03/2002 - 11/2002
CPR Infineon AG, Munich
- Diploma thesis: "A Computational Model for Sound Processing in the Human Auditory System"
- Supervisors: Dr. Werner Hemmert, Prof. Klaus Obermayer, Prof. Heinrich Klar

Working student, 08/2000 - 04/2001
ICN Siemens AG, Berlin
- Programming the test simulators for real time telephone systems
- Converting the Winhelp Systems into the Javahelp Systems

Internship Experience

Ph.D. Internship, 02/2007 - 05/2007
IBM T. J. Watson Research Center, Yorktown Heights, USA
- Research topic: Phrase extraction, domain adaptation for statistical machine translation
- Practical work: Building a large-scale Chinese-English statistical machine translation system

- Manager: Yuqing Gao, Mentor: Yonggang Deng
- Publications during this internship:

  J. Xu, Y. Deng, Y. Gao and H. Ney: Domain Dependent Machine Translation. In Proceedings of the Machine Translation Summit XI , Copenhagen, Danmark, September 2007.

  Y. Deng, J. Xu and Y. Gao: Phrase Table Training for Precision and Recall: What Makes a Good Phrase and a Good Phrase Pair? In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies , Columbus, OH, June 2008.

Ph.D. Internship, 10/2007 - 01/2008

Microsoft Research NLP group, Redmond, USA

- Research topic: Chinese word segmentation for statistical machine translation
- Practical work: NIST MT evaluation campaign 2007
- Manager: Bill Dolan, Mentor: Jianfeng Gao
- Publication during this internship:

  J. Xu, J. Gao, K. Toutanova and H. Ney: Bayesian Semi-Supervised Chinese Word Segmentation for Statistical Machine Translation. In Proceedings of the 22nd International Conference on Computational Linguistics , Manchester, UK, August 2008.
- Patent during this internship:

  Jianfeng Gao, Kristina Nikolova Toutanova and Jia Xu: Unsupervised Chinese Word Segmentation for Statistical Machine Translation. Redmond, USA, 27, June 2008.

Skills

- Operating Systems: Linux, Unix, Windows
- Programming Languages: C++, Java, Bash shell script, Makefile, Matlab
- Programming Environments: CVS, XEmacs, GDB
- Human languages: Chinese (native), English (fluent), German (fluent)

List of Publications

- J. Xu, J. Gao, K. Toutanova and H. Ney: Bayesian Semi-Supervised Chinese Word Segmentation for Statistical Machine Translation. In Proceedings of the 22nd International Conference on Computational Linguistics , Manchester, UK, August 2008.
- Y. Deng, J. Xu and Y. Gao: Phrase Table Training for Precision and Recall: What Makes a Good Phrase and a Good Phrase Pair? In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies , Columbus, OH, June 2008.
- J. Xu, Y. Deng, Y. Gao and H. Ney: Domain Dependent Machine Translation. In Proceedings of the Machine Translation Summit XI , Copenhagen, Danmark, September 2007.
- J. Xu, R. Zens and H. Ney: Partitioning Parallel Documents Using Binary Segmentation. In Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation, pp. 78-85, New York City, NY, June 2006.
- D. Vilar, J. Xu, L. F. D'Haro and H. Ney, Error Analysis of Statistical Machine Translation Output. In Proceedings of LREC 2006, pp. 697-702, Genova, Italy, May 2006.
- J. Xu, E. Matusov, R. Zens, and H. Ney: Integrated Chinese Word Segmentation in Statistical Machine Translation. In Proceedings of International Workshop on Spoken Language Translation, pp.141-147, Pittsburgh, PA, October, 2005.
- R. Zens, S. Hasan, J. Xu, Y. Zhang, H. Ney: The RWTH Statistical Machine Translation System for Chinese-English. NIST MT Evaluation Workshop, Washington D.C., June, 2005, Slides Only.
- R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, and H. Ney: The RWTH Phrase-based Statistical Machine Translation System. In Proceedings of the International Workshop on Spoken Language Translation (IWSLT), pp. 155-162, Pittsburgh, PA, October 2005.
- J. Xu, R.Zens, and H. Ney: Sentence Segmentation Using IBM Word Alignment Model 1. In Proceedings of the 10th Annual Conference of the European Association for Machine Translation (EAMT), pp. 280-287, Budapest, Hungary, May 2005.
- J. Xu, R. Zens, and H. Ney: Do We Need Chinese Word Segmentation for Statistical Machine Translation? In Proceedings of the Third SIGHAN Workshop on Chinese Language Learning, pp. 122-128, Barcelona, Spain, July 2004.
- R. Zens, O. Bender, S. Hasan, N. Ueffing, J. Xu, Y. Zhang, H. Ney: The RWTH Statistical Machine Translation System for Chinese-English. NIST MT Evaluation Workshop, Washington D. C., June 2004, Slides Only.
- J. Xu: A Computer Model for Sound Processing in the Human Auditory System. Diploma dissertation, Berlin, Germany, February 2003.

US Patent

- Jianfeng Gao, Kristina Nikolova Toutanova and Jia Xu: Unsupervised Chinese Word Segmentation for Statistical Machine Translation. Redmond, USA, 27, June 2008.