IMPROVING CHINESE-ENGLISH MACHINE TRANSLATION THROUGH BETTER SOURCE-SIDE LINGUISTIC PROCESSING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Pi-Chuan Chang
August 2009

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Christopher D. Manning) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Daniel Jurafsky)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Andrew Y. Ng)

Approved for the University Committee on Graduate Studies.

Abstract

Machine Translation (MT) is a task with multiple components, each of which can be very challenging. This thesis focuses on a difficult language pair – Chinese to English – and works on several language-specific aspects that make translation more difficult.

The first challenge this thesis focuses on is the differences in the writing systems. In Chinese there are no explicit boundaries between words, and even the definition of a "word" is unclear. We build a general purpose Chinese word segmenter with linguistically inspired features that performs very well on the SIGHAN 2005 bakeoff data. Then we study how Chinese word segmenter performance is related to MT performance, and provide a way to tune the "word" unit in Chinese so that it can better match up with the English word granularity, and therefore improve MT performance.

The second challenge we address is different word order between Chinese and English. We first perform error analysis on three state-of-the-art MT systems to see what the most prominent problems are, especially how different word orders cause translation errors. According to our findings, we propose two solutions to improve Chinese-to-English MT systems.

First, word reordering, especially over longer distances, caused many errors. Even though Chinese and English are both Subject-Verb-Object (SVO) languages, they usually use different word orders in noun phrases, prepositional phrases, etc. Many of these different word orders can be long distance reorderings and cause difficulty for MT systems. There have been many previous studies on this. In this thesis, we introduce a richer set of Chinese grammatical relations that describes more semantically abstract relations between words. We are able to integrate these Chinese grammatical relations into the most used, state-of-the-art phrase-based MT system and to improve its performance.

Second, we study the behavior of the most common Chinese word "的" (DE), which does not have a direct mapping to English. DE serves different functions in Chinese, and therefore can be ambiguous when translating to English. It might also cause longer distance reordering when translating to English. We propose a classifier to disambiguate DEs in Chinese text. Using this classifier, we improve the English translation quality because we can make the Chinese word orders much more similar to English, and we also disambiguate when a DE should be translated to different constructions (e.g., relative clause, prepositional phrase, etc.).

Acknowledgments

First, I would like to thank my advisor, Chris Manning, for being a great advisor in every way. On research, Chris has always provided very constructive comments during our weekly meetings. Also, Chris always offers good advice on writing and presenting my research work. He helps me organize the content of my papers and slides, and even fixes grammatical errors. If there are still any errors in this thesis, the original draft probably had 100 times more! I have enjoyed meeting with Chris, for he is very knowledgeable in various research topics, whether they are computer science or linguistics related.

I would like to thank Dan Jurafsky for his insightful ideas and suggestions during our collaboration on the DE classification and the grammatical relations work. In particular I want to thank him for his useful feedback to my research at MT meetings. I also would like to thank Andrew Ng, Peng Xu, and Yinyu Ye for being on my thesis committee and for their refreshing interest in my topic.

My dissertation topic is related to machine translation. On this part I want to give thanks to William Morgan, who sparked the initial interest in MT within the Stanford NLP group. I want to thank Kristina Toutanova for working with me on an MT project at Microsoft Research, which made me decide to work on MT for my dissertation. And also I would like to thank Michel Galley and Dan Cer in the MT group (and also my officemates) for useful discussions and collaborations on my research projects and the MT framework here at Stanford. I found that a good code-base and great people to work with are especially important when working on MT research. There is only so much that I can do by myself. I wouldn't be able to finish my dissertation without the help of the whole MT group.

Finally, I want to give thanks to the whole Stanford NLP group. It was always fun to talk to people at the weekly NLP lunch about what they are working on and what is going

on in life. During my years at Stanford, every member of the Stanford NLP group has been friendly and willing to help each other whenever anyone has questions. I greatly appreciate it.

Outside research and outside the department, I want to thank Hsing-Chen Tsai for being a good friend and supporting me through many difficult times during these years. And I want to thank Andrew Carroll for his support and love, and for sharing my happiness and sadness all the time. I would also like to thank my family in Taiwan – my dad, my mom and my sister, for supporting my decision to study abroad, and always praying for me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine translation (MT) has a long history. Early proposals for translating languages using computers were based on information theory, which was also used for code breaking in World War II. The area of machine translation was particularly interesting to the US government, usually in the direction of translating foreign languages to English. In the early days, Russian was one of the languages of interest, and currently Arabic and Chinese are the main source languages.

MT technology makes it possible and easier to collect knowledge in other languages, as well as to distribute knowledge to other languages. For commercial use, companies are more interested in translating from English to foreign languages. For example, Microsoft used MT technology to translate their technical manuals from English to many other languages. For personal use, there are also several MT systems online, such as Altavista's Babelfish (powered by the SYSTRAN system) and Google Translate.

Machine translation is a broad field with many different components in complicated systems. Some work focuses more on general language-independent techniques that can be applied to any language pair. For example, different formalisms of modeling the translation process, such as the commonly used phrase-based system (Moses) (Koehn et al., 2003) which is an instance of the noisy channel approach (Brown et al., 1993), and Hiero (Chiang, 2005) which is formally a synchronous context-free grammar (Lewis and Stearns, 1968).

Some other work focuses on more language-specific aspects, identifying and tackling difficulties in different language pairs. There are also many language-specific aspects that

researchers have worked on. For example, translation to morphologically complex languages can benefit from morphology generation models combined with the base MT system (Toutanova et al., 2008; Avramidis and Koehn, 2008). Translation between language pairs with very different word orders can be improved by reordering words in the source language to be closer to the target language, which requires understanding of the source and target language (Collins et al., 2005; Wang et al., 2007; Badr et al., 2009).

However, even the performance of language independent techniques can still be language-dependent. Birch et al. (2009) reported that synchronous grammar-based models produced superior results for Chinese-to-English MT, but for Arabic-to-English MT, phrase-based models are still competitive. This is because the Arabic-English pair has shorter range reordering where Moses performs better than Hiero, because Moses searches all possible permutations within a certain window whereas Hiero will only permit reorderings for which there is lexical evidence in the training corpus. The Chinese-English translation pair has more mid-range reordering where Hiero is better because its search space allows more longer distance reordering possibilities.

Therefore, we believe that language-specific knowledge is very critical for MT. Especially for languages that are more different from each other, we think that better understanding of the source language and the mapping of the source language to the target language is key for better MT quality. In Section 1.1, we will discuss some key issues in the Chinese language and why it can be difficult to translate to English. In Section 1.3 we will introduce phrase-based MT systems, which are the MT framework that we experiment on throughout the thesis. The main contributions of this thesis are described in Section 1.2. Each chapter of the thesis has relevant introductory materials and is self-contained.

1.1 Key issues in Chinese to English translation

Chinese and English differ in many ways. An obvious difference between them is their appearances (writing systems). The English alphabet is a Latin-based alphabet consisting of 26 letters, whereas the Chinese writing system contains tens of thousands of characters. According to the work by Chen et al. (1993), the most frequent 2452 characters and 28124 words make up 99% of the corpus they studied. On the word level, English words

are more well defined because there are spaces between words in a sentence, whereas in Chinese there are no delimiters between characters to indicate clear word boundaries, and the definition of a "word" in Chinese is debatable. Theoretical linguists have made some effort in defining words and have not come to a full agreement. On a more applied level, it has also been shown that different natural language processing (NLP) applications, such as information retrieval or speech recognition, would benefit from different segmentation standards (Peng et al., 2002; Gao et al., 2005). Also, because of the lack of similarity of characters and writing systems in general, certain techniques of recognizing similar words like cognates cannot be applied between these two languages.

In addition to the writing system difference, Chinese also uses less morphology and function words than English. Determiners for nouns are not required, nor plural marking of nouns. These differences have caused Chinese native speakers who learn English as a second language to make common mistakes like wrong tense agreement, not using plural nouns when necessary, incorrect usage of determiners (e.g., not adding determiners when needed, or inability to choose between definite or indefinite determiners), and they also have problems choosing among prepositions (Lee and Roukos, 2004; Turner and Charniak, 2007; Chodorow et al., 2007; Lee and Seneff, 2008). Also, Chinese verbs and nouns often have exactly the same surface form, whereas English words usually have different forms of nouns and verbs. For example, in English "invest" is the verb form of the noun "investment"; in Chinese "投资" is both the verb and noun form. Since this information is absent or implicit in Chinese, these differences have also become obstacles for computers translating from Chinese to English.

One other prominent difference is related to discourse and syntax. In Chinese it is very common to drop the subject of a sentence – the null realization of uncontrolled pronominal subjects, whereas in English pronominal subjects are usually required to produce a syntactically well-formed English sentence. This prevents current MT systems, which translate sentence by sentence and do not use discourse information, from recovering what the subject is, and thus the result is an incomplete English sentence structure.

Chinese and English have many major syntactic differences even though they are both Subject-Verb-Object (SVO) languages. First, prepositions in English occur before noun phrases (*pre*-position), such as "in the yard" and "on the table". In Chinese, there are also

postpositions, which occur after noun phrases. For example, “墙上有钉子” means “there is a nail on the wall”, where “上” is a postposition which means “on” and “墙” means “wall”. In the Chinese Treebank tagging guidelines, the postpositions are called *localizers* (LC). Second, in English a prepositional phrase (PP) usually follows the verb phrase (VP) it modifies while in Chinese the ordering is the opposite. For example,

Chinese:	我	在	家	吃	西瓜
Gloss:	I	at (preposition; ZAI)	home	eat	watermelon

Translation: I ate a watermelon at home.

This example illustrates the reordering between PP and VP. In Chinese the prepositional phrase “在 家” (at home) occurs before the verb phrase “吃 西瓜” (ate a watermelon). This reordering is discussed in a previous paper by Wang et al. (2007). In fact, the prepositions in Chinese and English have even more differences. In English, prepositions are different from verbs. But in Chinese, prepositions can be considered verbs. In Chinese grammar (Yip and Rimmington, 1997), 在 (ZAI) in the above example can be considered as a *coverb*. The coverbal phrase usually comes after the subject and before the main verb, providing background information about the place, time, methods, etc., associated with the main verb. In addition to 在 (ZAI), common coverbs include 到 (DAO, ‘to’), 往 (WANG, ‘towards’), 用 (YONG, ‘with, using’). These words can function as verbs or coverbs (or prepositions) in different context. For example, in this Chinese sentence:

Chinese:	我	用	毛笔	写	字
Gloss:	I	with (preposition; YONG)	Chinese writing brush	write	word

Translation: I wrote with a Chinese writing brush.

The word 用 (YONG) is a coverb (or a preposition). However, in a similar but shorter sentence “我 用 毛笔” (I use a Chinese writing brush), the word 用 (YONG) is the main verb.

Another syntactic difference that may cause longer-distance reordering is how modifiers are ordered with respect to what they modify. The PP and VP ordering above is one example. For nouns, English usually has shorter modifiers (adjective, possessive) in front of the noun being modified, but has longer modifiers (relative clauses, prepositional phrases) following the noun being modified. For example, an example of a short modifier that pre-modifies:

Chinese:	他	的	花
Gloss:	he	's (DE)	flower
Translation:	his flower		

And here is an example of a longer modifier that is a relative clause in English:

Chinese:	他	在	花店	买	的	花
Gloss:	he	in (preposition; ZAI)	flower shop	buy	(DE)	flower
Translation:	the flower he bought at the flower shop					

Note in both examples that there is the Chinese character 的 (DE) that doesn't always translate directly into English. In fact, the Chinese character 的 (DE) is marked with different POS tags and is translated differently in these two cases. 的 (DE) is also the most common Chinese word. This makes it important to disambiguate its role in the sentence and to understand whether reordering occurs. This is the main topic in Chapter 5.

Moreover, there are many Chinese syntactic constructions that are completely unseen in English. For example, the 把 (BA) and 被 (BEI) constructions are different, and also well discussed in the Chinese linguistics literature. The BA and BEI constructions have the object before the verb; this makes them an exception to the SVO structure in Chinese. In addition, although verbs in Chinese do not encode tense or finiteness features, they do inflect for aspect, including the perfective marker 了 (*-le*), the imperfective marker 着 (*-zhe*), and the experiential marker 过 (*-guo*).

Another phenomenon that can cause syntactic structural differences is *topicalization*. Li and Thompson (1976) described a language typology based on the grammatical relations *subject-predicate* and *topic-comment*. Since Chinese is a topic-prominent language, and English is a subject-prominent language, the word order can be very different when topicalization occurs. Topicalization in Chinese can move the topic of the current discourse to the beginning of a sentence. The topic always depends on an element inside the comment for its thematic role in the sentence, and it does not have its own syntactic function. Therefore the syntactic ordering of SVO might not apply here. For example, in the error analysis in Chapter 3, the first part of Sentence 38:

Chinese:	河	的	两	头	我	都	看	不	见
Gloss:	river	(DE)	two	end	I	both	look	no	see
Translation:	I could not see either end of the river								

In this example, the topic "河的两头" (either end of the river) is the object of the comment following it. So the Chinese sentence looks like an OSV structure, while the English translation is still an SVO structure.

In addition to the differences between Chinese and English, one major problem of Chinese-English MT is the lack of a careful error analysis of the state-of-the-art systems. In Chapter 3 we analyze three state of the art systems. In our analysis we showed examples of the above differences causing translation errors. For example, wrong and inconsistent segmentation makes it hard for the MT pipeline to recover from segmentation errors; certain Chinese constructions that require longer reordering tend to cause errors, etc.

1.2 Contributions

In this thesis, we address several issues that make Chinese to English machine translation difficult. In Chapter 2, we introduce a Chinese word segmenter that uses several linguistically inspired features to achieve good overall performance as well as high recall rate on out-of-vocabulary words. Our segmenter performed very well on all segmentation standards used in the SIGHAN bakeoff 2005. Since Chinese word segmentation is a first step in a lot of Chinese natural language processing (NLP) tasks, we released the Chinese segmenter[1] and it has been used by many other researchers. We also study the desirable characteristics for a segmenter to perform well on the Chinese to English MT task. We find that it is important to have consistent segmentation, and to use word units that are closer to English ones. We improve our segmenter by adding lexical features to improve consistency, and by tuning its word granularity to better match English word granularity.

Chapter 3 of the thesis includes error analysis of three state-of-the-art MT systems. It is important to observe the current systems to identify what are the most salient problems we are facing going forward. From our analysis, we choose two important directions of using Chinese linguistic structures to improve Chinese to English translation and study how to incorporate the linguistic information into a state-of-the-art phrase-based MT system.

In Chapter 4, we introduce a set of Chinese grammatical relations that can represent the

[1] http://www-nlp.stanford.edu/software/segmenter.shtml

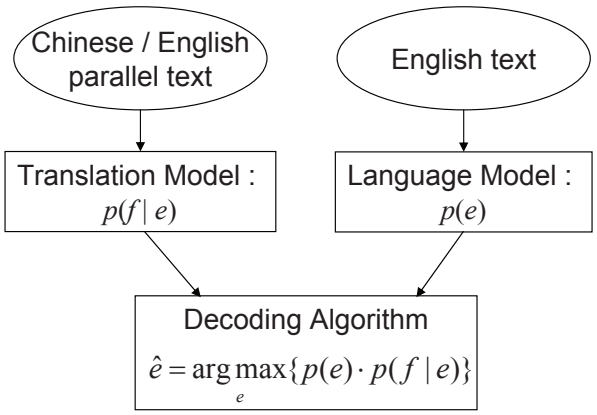


Figure 1.1: Architecture of the translation approach based on the noisy channel model.

sentence structure and semantic relations between words. We use these grammatical relations to design a feature for phrase-based MT systems that can help in reordering decision and thus improve MT performance. We also hope that this new set of Chinese grammatical relations can be used in many other Chinese NLP tasks, especially ones related to meaning extraction. The grammatical relations are also released to the public as part of the Stanford Parser.[2]

In Chapter 5, we focus on the most common word "的" (DE) in Chinese that can be ambiguously translated, which has caused serious issues in word reordering in translation. More specifically, 的 can cause longer range reordering, which is difficult for phrase-based systems and even hierarchical systems (Birch et al., 2009). Therefore, we propose to build a DE classifier that disambiguates DEs according to how they are translated to English, and show that using this DE classifier can improve MT system performance significantly.

1.3 Background: Phrase-based MT Systems

Phrase-based systems are currently the most commonly used statistical MT systems. Moses, an open source phrase-based system, is specially popular among the MT research community. In this work, we use *Phrasal*, an MT decoder written by Dan Cer, which is similar at a high level, but allows new features to be added more easily. In this section we will give

[2] http://www-nlp.stanford.edu/software/lex-parser.shtml

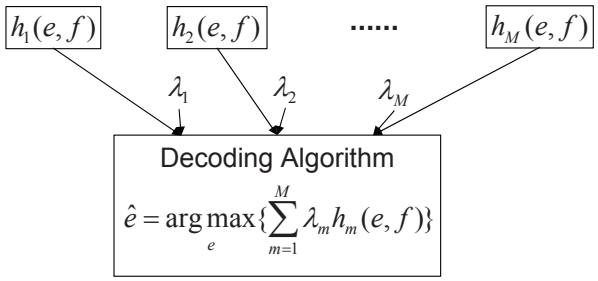


Figure 1.2: Architecture of the translation approach based on log-linear models.

a overall sketch of what components there are in a phrase-based MT system, and we will emphasize the components that are more relevant to the thesis. Since the thesis is about Chinese to English translation, instead of using general terminology like "source" and "target", or "foreign" and "English", we will use *Chinese* to refer to the source language and *English* to refer to the target language. But we will still use the symbol f to refer to Chinese (foreign) sentences and e for English ones.

The translation task is when we are given a Chinese sentence f and want to translate it into a English sentence e. The phrase-based framework is based on the noisy channel model. We can use Bayes rule to reformulate the translation probability:

$$\hat{e} = \arg\max_e p(e|f) = \arg\max_e p(e)p(f|e)$$

Using this decomposition, the system can have a language model $p(e)$ and a translation model $p(f|e)$, as depicted in Figure 1.1.

This is the basic concept of current statistical machine translation (SMT) systems, including phrase-based and hierarchical systems. In real implementation, a more general log-linear framework (Och and Ney, 2002) is used instead of just a simple $p(e)p(f|e)$ decomposition. The log-linear framework is illustrated in Figure 1.2. The general log-linear framework is the idea of having arbitrary feature functions $h_m(e,f)$ used in the decoding algorithm, weighted by weights λ_m.

The simple noisy channel model in Figure 1.1 is actually a special case of the general log-linear framework in Figure 1.2, where $M = 2$ and the two feature functions and weights

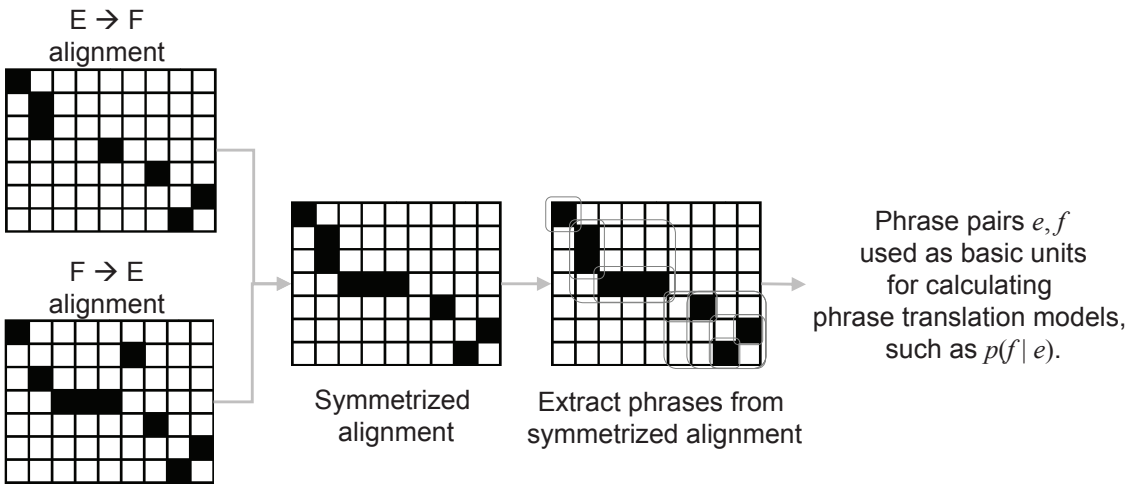


Figure 1.3: Phrase extraction pipeline.

are:

$$h_1(e,f) = \log p(f|e)$$
$$h_2(e,f) = \log p(e)$$
$$\lambda_1 = 1$$
$$\lambda_2 = 1$$

The advantage of the log-linear framework is that the weights (λ_i) for feature functions can be tuned to optimize for performance. The most commonly used technique for tuning is Minimum Error Rate Training (MERT), introduced by Och (2003), which can directly optimize weights for translation quality.

A distinctive characteristic of phrase-based MT systems is how the phrases are defined and extracted from parallel text. The technique of using extracted statistical phrases has proven very successful in SMT systems, and also helps explain one of the hypotheses (Hypothesis 1) in Chapter 2. So we will introduce phrase extraction algorithms in Section 1.3.1 in more detail.

1.3.1 Phrase extraction

One important improvement from earlier word-based MT systems (Brown et al., 1993) to phrase-based systems is the use of *phrases* (chunks of words) as basic translation units. The

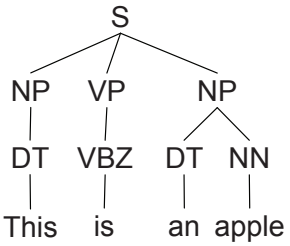


Figure 1.4: Parse tree for the sentence "This is an apple".

phrases in phrase-based systems are not *linguistic* phrases as in constituents of parse trees, but are *statistical* phrases extracted from symmetrized word alignments in bi-text (Och and Ney, 2003). For example, the English sentence "This is an apple" has a parse tree shown in Figure 1.4. A linguistic phrase is like "an apple", which is a noun phrase in the parse tree. The fragment "This is" is not a linguistic phrase, since there is not any subtree that spans only the words "This is". However in a phrase-based system, it is possible that the system extract "This is" as a *statistical* phrase. The use of phrases allows phrase-based systems to handle one-to-many, many-to-one, or even many-to-many translation. For Chinese to English translations, the alignments and phrases can even be used to find suitable chunks of Chinese characters for machine translation (Xu et al., 2004). This is related to how different Chinese segmentation strategies work with phrase-based MT systems, which is important to Chapter 2.

To extract phrases, the pipeline is illustrated in Figure 1.3. First, IBM-style word alignments (Brown et al., 1993) are run in both directions, which generate many-to-one and one-to-many word alignments. After bidirectional alignments are generated, there are different heuristics to merge them into one symmetrized alignment (Och and Ney, 2003). The default heuristic used in Moses is grow-diag-final, which applies grow-diag and final. grow-diag starts with the intersection of the two IBM-style alignments – only word alignment points that occur in both alignments are preserved. In the second step of grow-diag, additional alignment points are added. Only alignment points that are in either of the two IBM-style alignments are considered. In this step, potential alignment points that connect currently unaligned words and that neighbor established alignment points are added. Neighboring are defined as the left, right, top, bottom, or diagonal alignment points. After this step, the resulting symmetrized alignment is grow-diag. In the final step, alignment

points in either of the two alignments that do not neighbor established alignment points are added.

In our experience, we found that using grow-diag gives better Chinese-English MT performance. Regardless of which merging heuristic is used, phrases can be extracted from the merged alignments. The words in a legal phrase pair are only aligned to each other, and not to words outside (Och et al., 1999) This phrase extraction algorithm generates a sparser word alignment, which allows extraction of more phrases. For example, the sparser word alignment of grow-diag extracts more phrases than grow-diag-final.

1.3.2 Basic feature functions in MERT

In Moses, there are 9 basic feature functions, including 5 features from the phrase table, one linear distortion model, one unknown word feature, one N-gram language model feature, and one word penalty feature. However, in the Moses implementation, the weight for the unknown word feature is not tuned. In *Phrasal*, all of the 9 basic feature functions are tuned in MERT. In addition to these 9 features, lexicalized reordering models[3] have also become a common baseline that people compare to. Moses and *Phrasal* both have implementations of lexicalized reordering models. Adding on the lexicalized reordering models increases the number of MERT features by 6. In this thesis, a common baseline is with 15 features. In Chapter 4 we also have a baseline without the lexicalized features (9 features).

[3] http://www.statmt.org/moses/?n=Moses.AdvancedFeatures#ntoc1

Chapter 2

Chinese Word Segmentation and MT

This chapter introduces the Chinese word segmentation problem, which is a fundamental first step of Chinese NLP tasks. In this chapter we discuss our design of a segmenter that performs very well on general Chinese word segmentation, using linguistically inspired features. We also discuss the impact of Chinese word segmentation on a statistical MT system, and further improve the segmenter specifically for the MT task.

2.1 Chinese Word Segmentation

Word segmentation is considered an important first step for Chinese natural language processing tasks, because Chinese words can be composed of multiple characters with no space appearing between words. Almost all tasks could be expected to benefit by treating the character sequence "天花" together, with the meaning *smallpox*, rather than dealing with the individual characters "天" (*sky*) and "花" (*flower*). Without a standardized notion of a word, the task of Chinese word segmentation traditionally starts from designing a segmentation standard based on linguistic and task intuitions, and then aiming to build segmenters that output words that conform to the standard. An example of multiple segmentation standards is in the SIGHAN bakeoffs for Chinese word segmentation, where there are several corpora segmented to different standards. For example, SIGHAN bakeoff 2005 provided four different corpora from Academia Sinica, City University of Hong Kong, Peking University, and Microsoft Research Asia. Other than these, one widely used

standard is the Penn Chinese Treebank (CTB) Segmentation Standard (Xue et al., 2005).[1]

In this chapter, we start by formally defining the segmentation task, and introduce a simple and commonly-used segmentation paradigm, *lexicon-based*, in Section 2.1.1. Then in Section 2.2 we describe the *feature-based* approach. We experiment with different features to build a segmenter that performs very well across all the segmentation standards. The reason why the features we use are robust across segmentation standards is because most differences among standards result from morphological processes. As observed in (Wu, 2003), if we compare the various standards, there are more similarities than differences. The differences usually involve words that are not typically in the dictionary. Wu (2003) called those words morphologically derived words (MDWs) because they are more dynamic and usually formed through productive morphological processes. These words occur where different standards usually have different segmentation decisions. Wu (2003) discussed several morphological processes such as reduplication, affixation, directional and resultative compounding, merging and splitting, and named entities and factoids. These morphological processes inspired the feature design of the segmenter described in Section 2.2, which is feature-based and the weights of each feature can be learned on different corpora to mimic how various standards make different decisions on whether to split a word or not.

In addition to multiple segmentation standards, another complication of Chinese word segmentation comes from the fact that different applications require different granularities of segmentation. In particular, we want to understand how to improve segmentation for Chinese to English machine translation systems. In Section 2.3, we have more discussion and experiments on how Chinese word segmentation affects MT systems, and also introduce an improved segmenter that combines the benefits of both the feature-based and lexicon-based paradigms, and adjusts for optimal MT performance.

2.1.1 Lexicon-based Segmenter

Given a sentence of n Chinese characters $\mathsf{S}= c_1c_2...c_n$, S can be segmented into m non-overlapping adjacent substrings $\mathsf{G} = w_1, w_2, ..., w_m$, where every substring $w_i = c_p...c_q$ is a

[1] This chapter includes joint work with many colleagues; mainly from the two papers (Tseng et al., 2005) and (Chang et al., 2008).

word and G is called a segmentation. In the example in Figure 2.1, segmentation G_0 is the *trivial* segmentation, where every character is regarded as an individual word. Later in this chapter, the trivial segmentation is also referred to as the *character-based* segmentation. To formalize the problem, we can associate each character with a label 0 or 1 to indicate if there is a word boundary *before* the current character. The example in Figure 2.1 is a sentence S="斯坦福大学" (Stanford University) The set of all feasible segmentations is $\mathscr{G}$(S)={G_0, ..., G_k}. For example, G_0 in Figure 2.1 will have the label sequence $L_0 = 11111$, and G_k will have the label sequence $L_k = 10010$. With this definition, there are 2^{n-1} possible label sequences (because the first character always has the label 1).

With lexicon-based approaches, there exists a lexicon to start with. For the example in Figure 2.1, if the lexicon is {"斯", "坦", "福", "大", "学", "斯坦福", "大学"}, the possible label sequences will be constrained from $2^4 = 16$ to 4 : 11111, 11110, 10010, and 10011. If the lexicon contains every character as a single-character word, the trivial segmentation is one of the possible segmentations. Lexicon-based approaches include the simplest forward maximum matching (Wong and Chan, 1996), that looks for the longest matched lexicon word in from left to right. A variant is backward maximum matching, which matches the longest word from right-to-left instead of left-to-right. Both forward and backward maximum matching are *greedy* algorithms, Every step they look for the longest matched word, which makes the segmentation decision fast, but might not be optimal. Among the lexicon-based approaches, there are also more sophisticated ones, such as using an n-gram language model to define the objective function. If we have a gold segmented Chinese text, and train an n-gram language model on it, we can use the language model to score the log-likelihood of a particular segmentation G:

$$L(\mathsf{G}) = \log P(\mathsf{G}) = \sum_{i=1}^{m} \log P(w_i | w_{i-n+1} ... w_{i-1})$$

And the best segmentation will be the one with the highest log-likelihood:

$$\mathsf{G}^* = \arg\max_{G} L(\mathsf{G})$$

Finding the best segmentation can be done efficiently by dynamic programming.

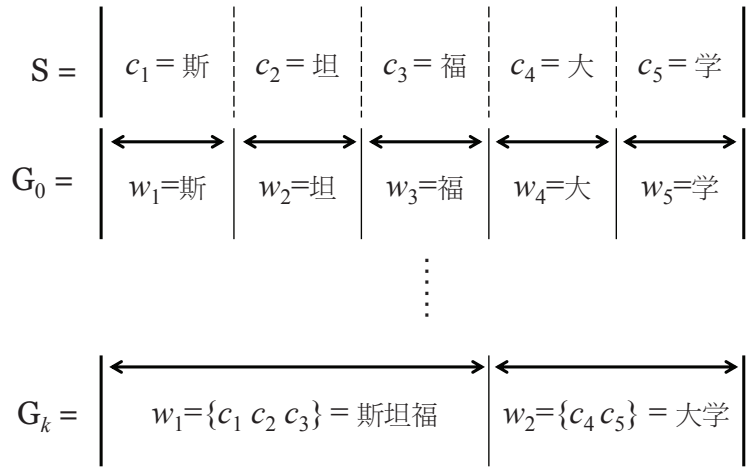


Figure 2.1: A Chinese sentence S with 5 characters. $\mathscr{G}(S)=\{G_0, \ldots, G_k\}$ is the set of possible segmentations.

Even with the shortcoming that out-of-vocabulary words cannot be detected, lexicon-based approaches still remain a very common segmentation technique for many applications or as a baseline, especially the forward maximum matching technique, because it only requires a pre-defined lexicon and does not require any extra statistical information.

2.2 Feature-based Chinese Word Segmenter

This section describes the feature-based segmenter inspired by the morphological processes that generate Chinese words. The segmenter builds on a conditional random field (CRF) framework which makes it easier to integrate various linguistic features and make a global segmentation based on what features are present in the context.

Compared to the lexicon-based approaches in Section 2.1.1, the search space of a CRF segmenter is not constrained by a lexicon, therefore it has the ability to recognize unseen new words in context, and takes more linguistic features into account.

2.2.1 Conditional Random Field

Conditional random fields is a statistical sequence modeling framework first introduced by Lafferty et al. (2001). Work by Peng et al. (2004) first used this framework for Chinese

word segmentation by treating it as a binary decision task, such that each character is labeled either as the beginning of a word or the continuation of one. The probability assigned to a label sequence for a particular sequence of characters by a first-order CRF is given by the equation below:

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{x}, y_{t-1}, y_t, t) \tag{2.1}$$

$\mathbf{x}$ is a sequence of T unsegmented characters, $Z(\mathbf{x})$ is the partition function that ensures that Equation 2.1 is a probability distribution, $\{f_k\}_{k=1}^{K}$ is a set of feature functions, and $\mathbf{y}$ is the sequence of binary predictions for the sentence, where the prediction $y_t = 1$ indicates the t-th character of the sequence is preceded by a space, and where $y_t = 0$ indicates there is none. Our Chinese segmenter uses the CRF implementation by Jenny Finkel (Finkel et al., 2005). We optimized the parameters with a quasi-Newton method, and used Gaussian priors to prevent overfitting.

2.2.2 Feature Engineering

The linguistic features used in the model fall into three categories:

1. character identity n-grams
2. morphological features
3. character reduplication features

The first category, character identity features, has been used in several Chinese sequence modeling papers such as the joint word segmentation and part-of-speech (POS) tagging work of Ng and Low (2004), and the segmentation work of Xue and Shen (2003). Character identity features turn out to be a basic feature that people use despite the differences of their approaches.

Our character identity features are represented using feature functions that key off of the identity of the characters in the current, preceding, and subsequent positions. Specifically, we used four types of unigram feature functions, designated as C_0 (current character),

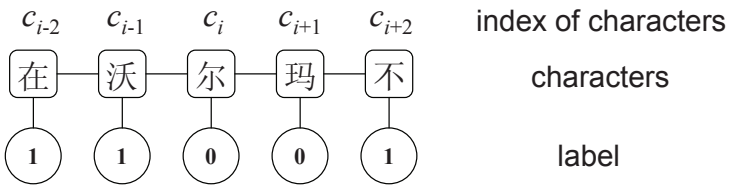


Figure 2.2: An example of a five-character sequence in a Chinese sentence. Label 1 means there is a boundary in front of the character, and label 0 means the character is a continuation of the previous character.

C_1 (next character), C_{-1} (previous character), C_{-2} (the character before the previous character). Other than the single character identity features, five types of bigram features were used, and are notationally designated here as conjunctions of the previously specified unigram features, C_0C_1, $C_{-1}C_0$, $C_{-1}C_1$, $C_{-2}C_{-1}$, and C_0C_2. Figure 2.2 is an example of a fragment of five Chinese characters in a sentence. If the current position is at c_i in the figure, the features we will be getting are: unigram features: C_0-尔, C_1-玛, C_{-1}-沃, C_{-2}-在, and bigram features: C_0C_1-尔玛, $C_{-1}C_0$-沃尔, $C_{-1}C_1$-沃玛, $C_{-2}C_{-1}$-在沃, and C_0C_2-尔不. Note that since the label for C_0 is deciding the boundary *in front of* the character C_0, using features from C_{-2}, C_{-1}, C_0, C_1 is actually taking a symmetric window of 2 from both sides of the boundary.

Since the difficulty in Chinese word segmentation often results from words that are not in the dictionary, we also defined several morphologically inspired features to help recognize those unknown words. Given that unknown words are normally more than one character long, when representing the morphological features as feature functions, such feature functions keyed off the morphological information extracted from both the preceding state and the current state. We have three types of morphological features, based upon the intuition regarding unknown word features given in (Gao et al., 2004). Specifically, the idea was to use productive affixes and characters that only occurred independently to predict boundaries of unknown words. Our morphological features include:

1. Prefix and Suffix characters of unknown words
2. Stand-alone single-character words
3. Bi-characters at word boundaries

For morphological feature 1, in order to comply with the rules in the closed track of SIGHAN bakeoffs, we construct a table containing affixes of unknown words by extracting rare words from the corpus, and then collect the first and last characters from them to construct the prefix and suffix character tables of unknown words. When extracting features, we put in a prefix feature if the character at the previous position (C_{-1}) is present in the prefix table, and put in a suffix feature if the character at the current position (C_0) is present in the suffix table.

For the table of individual character words (morphological feature 2), we made an individual character word table for each corpus by collecting characters that always occurred alone as a separate word in the given corpus. This table is used to matched the current, preceding or next character to extract features. For example, if the current position (C_0) in Figure 2.2 is c_{i-1}, and only the character "在" is in the table, the feature "SINGLECHAR-C_0-沃" and "SINGLECHAR-C_1-尔" will not be true, and only the feature "SINGLECHAR-C_{-1}-在" will be set to true.

We also collected a list of bi-characters from each training corpus to distinguish known strings from unknown (morphological feature 3.) This table is done by collecting bi-character sequences that occurred at the boundary of two subsequent words, and never occurred subsequently within a word. For example, for the two Chinese words "在(at) / 沃尔玛(Walmart)", we put in the table a bi-character entry "在沃" because these two characters had a boundary in between, and there are no words that contains the bi-character pattern "在沃" in them. Once we have the table, the word boundary bi-character feature fires when the bi-character sequence of previous word and the current word exist in the table. For example, in Figure 2.2 if the current position (C_0) is at c_{i-1}, since $C_{-1}C_0$ is "在沃" and it is in the table, the feature "UNK-在沃" will be set to true.

Additionally, we also use reduplication features that are active based on the repetition of a given character. (Wu, 2003) has an extensive discussion and examples of reduplication in Chinese. The main patterns of reduplication in Chinese are AA, ABAB, AABB, AXA, AXAY, XAYA, AAB and ABB. For example, "看看" (look-look "take a look") has the pattern AA, and "讨论讨论" (discuss-discuss "have a discussion") has the pattern ABAB. Since the meaning of AA and ABAB is not compositional, some standards considered both single words. However, some other standards decided to break "讨论讨论" because

Corpus	Abbrev.	Encoding	#Train. Words	#Test. Words
Academia Sinica	**AS**	Big Five (MS Codepage 950)	5.8M	12K
U. Penn Chinese Treebank	**CTB**	EUC-CN (GB2312-80)	250K	40K
Hong Kong CityU	**HK**	Big Five (HKSCS)	240K	35K
Beijing University	**PK**	GBK (MS Codepage 936)	1.1M	17K

Table 2.1: Corpus Information of SIGHAN Bakeoff 2003

"讨论" (discussion) itself can be looked up in the dictionary. We designed reduplication features so that the weights can be learned based on different standards. We have two reduplication feature functions, one fires if the previous and the current character (C_{-1} and C_0) are identical, and the other fires if the subsequent and the previous character (C_{-1} and C_1) are identical.

Adopting all the features together in a model and using the automatically generated morphological tables prevented our system from manually overfitting the Mandarin varieties we are most familiar with, and therefore enables our segmenter to work well on all of the segmentation standards we tested with.

Most features appeared in the first-order templates in the CRF framework with a few character identity features in both the zero-order and first-order templates. We also did punctuation normalization due to the fact that Mandarin has a huge variety of punctuations. The punctuations were extracted from the corpora and were all normalized into one single symbol to represent punctuations.

2.2.3 Experimental Results

We developed our segmenters on data sets from the First SIGHAN Chinese Word Segmentation Bakeoff (Sproat and Emerson, 2003), and tested on data sets from the Second SIGHAN Chinese Word Segmentation Bakeoff (Emerson, 2005).

The SIGHAN 2003 bakeoff provided different corpora from four different sites, containing different amount of training and testing data, and different encodings of Chinese characters. The corpora details of SIGHAN 2003 bakeoff are listed in Table 2.1.

Our system's F-scores on post-hoc testing on the SIGHAN 2003 corpora are reported in Table 2.2. Table 2.2 also includes numbers from the work by Peng et al. (2004) that also built a CRF segmenter and reported on SIGHAN 2003 data. From the table we can see that

SIGHAN Bakeoff 2003	Our F-score	Peng et al. (2004)	Bakeoff Best
AS	0.970	0.956	0.961
CTB	0.863	0.849	0.881
HK	0.947	0.928	0.940
PK	0.953	0.941	0.951

Table 2.2: Comparisons of (Peng et al., 2004), our F-scores, and the best bakeoff score on the closed track in SIGHAN bakeoff 2003 (Sproat and Emerson, 2003)

Corpus	Abbrev.	Encodings	Training Size (Words / Types)	Test Size (Words / Types)
Academia Sinica (Taipei)	**AS**	Big Five Plus, Unicode	5.45M / 141K	122K / 19K
Beijing University	**PK**	CP936, Unicode	1.1M / 55K	104K / 13K
Hong Kong CityU	**HK**	Big Five/HKSCS, Unicode	1.46M / 69K	41K / 9K
Microsoft Research (Beijing)	**MSR**	CP936, Unicode	2.37M / 88K	107K / 13K

Table 2.3: Corpus Information of SIGHAN Bakeoff 2005

our segmenter performs better over the strong baselines in (Peng et al., 2004). We attribute this to the morphologically inspired features we added to our segmenter.

In the SIGHAN 2005 bakeoff, there were also four corpora. But instead of **CTB**, they have a corpus from Microsoft Research Asia as the fourth corpus. The corpora statistics are listed in Table 2.3. Our final system achieved a F-score of 0.947 on **AS**, 0.943 on **HK**, 0.950 on **PK** and 0.964 on **MSR**, with the detailed breakdown of precision, recall, recall on out-of-vocabulary (OOV) words and recall on in-vocabulary (IV) words listed in Table 2.4. Our system participated in the closed division of the SIGHAN 2005 bakeoff and was ranked first in **HK**, **PK**, and **MSR** and tied with the Yahoo system on **AS** as the first place. When we compared the detailed performance with other systems, we observed that our recall rates on OOV words are much higher than other systems, and our recall rates on IV words are also usually higher than competing systems. This confirms the assumption that using morphologically inspired features is beneficial and learning weights for those features enables us to adapt to different segmentation standards.

2.2.4 Error Analysis

Our system performed reasonably well on morphologically complex new words, such as 电缆线 ("cable line" in **AS**) and 杀人案 ("murder case" in **PK**), where 线 (line) and 案

	R	P	F	R_{OOV}	R_{IV}
AS	0.950	0.943	0.947	0.718	0.960
PK	0.941	0.946	0.943	0.698	0.961
HK	0.946	0.954	0.950	0.787	0.956
MSR	0.962	0.966	0.964	0.717	0.968

Table 2.4: Detailed performances on SIGHAN bakeoff 2005. R: recall, P: precision, F: F-score, R_{OOV}: recall on out-of-vocabulary words, R_{IV}: recall on in-vocabulary words.

(case) are suffixes. However, it overgeneralized to wrongly propose words with frequent suffixes such as 灼人 (it should be 灼 / 人 "to burn someone" in **PK**) and 过头 (it should be 回过 / 头 "to look backward" in **PK**). For the corpora that considered 4 character idioms as a word, our system combined most of the new idioms together. This differs greatly from the results that one would likely obtain with a more traditional maximum matching based technique, as such an algorithm would segment novel idioms. The ability to generalize to recognize OOV words is a strength of our system, however it might lead to some other problems in applications such as machine translation, as described in Section 2.3.

Another common mistake of our system is that it is not able to distinguish a subtle semantic decision between ordinal numbers and numbers with measure nouns. The CTB segmentation standard makes different segmentation decisions for these two semantic meanings. For example, "the third year" and "three years" are both "三年" in Chinese. But in the CTB segmentation standard, "the third year" is one word: 三年, and "three years" is segmented into two words: 三 / 年. Our system is not able to distinguish between these two cases. In order to avoid this problem, it might require having more syntactic knowledge than was implicitly given in the training data. Finally, some errors are due to inconsistencies in the gold segmentation of non-hanzi characters. For example, "Pentium4" is a word, but "PC133" is two words. Sometimes, −8℃ is a word, but sometimes it is segmented into two words.

Overall, based on the performance reported on SIGHAN datasets (Table 2.2 and 2.4) and this error analysis, our segmenter is adaptive when training on different standards, and does a good job on recognizing OOV words (high recall rate on OOV words as shown in the tables). The segmenter performs well on the F-score measure commonly used in segmentation evaluation. In the following sections, we will talk about how it correlates to

performance in higher level applications such as MT.

2.3 Word Segmentation for Machine Translation

The importance of Chinese word segmentation as a first step for Chinese natural language processing tasks has been discussed in Section 2.1. The problem gets more complicated when different applications are considered, because it has been recognized that different applications have different needs for segmentation. Chinese information retrieval (IR) systems benefit from a segmentation that breaks compound words into shorter "words" (Peng et al., 2002), paralleling the IR gains from compound splitting in languages like German (Hollink et al., 2004), whereas automatic speech recognition (ASR) systems prefer having longer words in the speech lexicon (Gao et al., 2005).

However, despite a decade of very intense work on Chinese to English machine translation, the way in which Chinese word segmentation affects MT performance is very poorly understood. With current statistical phrase-based MT systems, one might hypothesize that segmenting into small chunks, including perhaps even working with individual characters would be optimal. This is because the role of a phrase table is to build domain and application appropriate larger chunks that are semantically coherent within the translation process. Hence the word segmentation problem can be circumvented to a certain degree, because the construction of the phrase table might be able to capture multiple characters forming a word. For example, even if the word for *smallpox* is treated as two one-character words, they can still appear in a phrase like "天 花→*smallpox*", so that *smallpox* will still be a candidate translation when the system translates "天" "花". Nevertheless, Xu et al. (2004) show that an MT system with a word segmenter outperforms a system working with individual characters in an alignment template approach. On different language pairs, Koehn and Knight (2003) and Habash and Sadat (2006) show that data-driven methods for splitting and preprocessing can improve Arabic-English and German-English MT.

Beyond this, there has been no finer-grained analysis of what style and size of word segmentation is optimal for MT. Moreover, most of the discussion of segmentation for other tasks relates to the size units identified in the segmentation standard: whether to join or split noun compounds, for instance. People generally assume that improvements

in a system's word segmentation accuracy will be monotonically reflected in overall system performance. This is the assumption that justifies the recent concerted work on the independent task of Chinese word segmentation evaluation at SIGHAN and other venues. However, we show that this assumption is false: aspects of segmenters other than error rate are more critical to their performance when embedded in an MT system. Unless these issues are attended to, simple baseline segmenters can be more effective inside an MT system than more complex machine learning based models with much lower word segmentation error rate.

In this section, we design several experiments to support our points. We will show that even having a basic word segmenter helps MT performance, and we analyze why building an MT system over individual characters (i.e., no word segmentation) doesn't function as well (Section 2.3.2). We also demonstrate that segmenter performance is not monotonically related to MT performance, and we analyze what characteristics of word segmenters most affect MT performance (Section 2.3.3). Based on an analysis of baseline MT results, we pin down four issues of word segmentation that can be improved to get better MT performance.

1. While a feature-based segmenter, like the one we described in Section 2.2, may have very good aggregate performance, inconsistent context-specific segmentation decisions can be harmful to MT system performance.

2. A perceived strength of feature-based systems is that they are able to generate out-of-vocabulary (OOV) words. as OOV words can hurt MT performance in cases when they could have been split into subparts from which the meaning of the whole can be roughly compositionally derived.

3. Conversely, splitting OOV words into non-compositional subparts can be very harmful to an MT system: it is better to produce such OOV items than to split them into unrelated character sequences that are known to the system. One big source of such OOV words is named entities.

4. Since the optimal granularity of words for phrase-based MT is unknown, we can benefit from a model which provides a knob for adjusting average word size.

We build several different models to address these issues and to improve segmentation for the benefit of MT. First, we extend the features in the segmenter from Section 2.2 to emphasize lexicon-based features in a feature-based sequence classifier to deal with segmentation inconsistency and over-generating OOV words. Having lexicon-based features reduced the MT training lexicon by 29.5%, reduced the MT test data OOV rate by 34.1%, and led to a 0.38 BLEU point gain on the test data (MT05). Second, we extend the CRF label set of our CRF segmenter to identify proper nouns. This gives 3.3% relative improvement on the OOV recall rate, and a 0.32 improvement in BLEU. Finally, in Section 2.3.4 and 2.3.5 we tune the CRF model to generate shorter or longer words to directly optimize the performance of MT. For MT, we found that it is preferred to have words slightly shorter than the CTB standard. We also incorporate an external lexicon and information about named entities for better MT performance.

2.3.1 Experimental Setting

Since we want to understand how segmenter performance is related to MT performance, we need to describe the experimental settings for both the Chinese word segmentation system and the machine translation system we are using.

Chinese Word Segmentation

For directly evaluating segmentation performance, we train each segmenter with the UPUC data set (University of Pennsylvania and University of Colorado, Boulder) of the SIGHAN Bakeoff 2006 training data and then evaluate on the test data. The reason why we chose this segmentation standard is that it is used in the most commonly used Chinese linguistic resources, such as the Chinese Treebank. The training data contains 509K words, and the test data has 155K words. The percentage of words in the test data that are unseen in the training data is 8.8%. Details of the Bakeoff data sets are in (Levow, 2006). To understand how each segmenter learns about OOV words, we will report the F-score, the in-vocabulary (IV) recall rate as well as OOV recall rate of each segmenter.

Phrase-based Chinese-to-English MT

As introduced in Chapter 1, the MT system we use is a re-implementation of Moses, a state-of-the-art phrase-based system (Koehn et al., 2003). We build phrase translations by first acquiring bidirectional GIZA++ (Och and Ney, 2003) alignments, and using Moses' grow-diag alignment symmetrization heuristic. As explained in Chapter 1, the grow-diag heuristic generates sparser alignments and therefore more phrases can be extracted. In our experiments, the grow-diag heuristic consistently performed better than the default (grow-diag-final) heuristic in Moses. We also extended the maximum phrase length from the default 7 to a larger value 10. Increasing the maximum phrase length allows better comparisons between segmenters, because some segmenters generate shorter words. During decoding, we incorporated the standard eight feature functions of Moses as well as the lexicalized reordering model. We tuned the parameters of these features with Minimum Error Rate Training (MERT) (Och, 2003) on the NIST MT03 Evaluation data set (919 sentences), and then tested the MT performance on NIST MT02 and MT05 Evaluation data (878 and 1082 sentences, respectively). We report the MT performance using the original BLEU metric (Papineni et al., 2001). The BLEU scores reported are uncased.

The MT training data was subsampled from the DARPA GALE program Year 2 training data using a collection of character 5-grams and smaller n-grams drawn from all segmentations of the test data. Since the MT training data is subsampled with character n-grams, it is not biased towards any particular word segmentation. The MT training data contains 1,140,693 sentence pairs; on the Chinese side there are 60,573,223 non-whitespace characters, and the English sentences have 40,629,997 words.

Our main source for training our five-gram language model was the English Gigaword corpus, and we also included close to one million English sentences taken from LDC parallel texts: GALE Year 1 training data (excluding FOUO data), Sinorama, Xinhua News, and Hong Kong News. We restricted the Gigaword corpus to a subsample of 25 million sentences, because of memory constraints.[2]

[2]We experimented with various subsets of the Gigaword corpus, and tested different trade-offs between using all the data included in our subset (at the cost of restricting 4-grams and 5-grams to the most frequent ones) and using specific subsets we deemed more effective (with the advantage that we could include all 4-grams and 5-grams that occur at least twice). Overall, we found that the best performing model was one trained with all the selected data, while restricting the set of 4-grams and 5-grams to those occurring at least

2.3.2 Understanding Chinese Word Segmentation for Phrase-based MT

In this section, we experiment with three types of segmenters – character-based, lexicon-based and feature-based – to explore what kind of characteristics are useful for segmentation for MT.

Character-based, Lexicon-based and Feature-based Segmenters

The supervised word segmentation training data available for the segmenter is two orders of magnitude smaller than the parallel data available for the MT system, and they are also not well matched in terms of genre and variety. Also, when training word alignment between Chinese and English, the information an MT system learns might be provides a basis of a more task-appropriate segmentation style for Chinese-English MT. A phrase-based MT system like Moses can extract "phrases" (sequences of tokens) from a word alignment and the system can construct the words that are useful. These observations suggest the first hypothesis.

Hypothesis 1. *A phrase table should capture word segmentation. Character-based segmentation for MT should not underperform a lexicon-based segmentation, and might outperform it.*

Observation In the experiments we conducted, we found that the phrase table cannot capture everything a Chinese word segmenter can do, and therefore having word segmentation helps phrase-based MT systems.[3]

To show that having word segmentation helps MT, we compare a lexicon-based maximum matching segmenter with character-based segmentation (treating each Chinese character as a word). We choose the most common maximum matching algorithm from Section 2.1.1 as our lexicon-based segmenter. We will later refer to this segmenter as MaxMatch.

five times in the training data.

[3]Different phrase extraction heuristics might affect the results. In our experiments, grow-diag outperforms both one-to-many and many-to-one for both MaxMatch and CharBased. We report the results only on grow-diag.

Segmentation Performance

Segmenter	F-score	OOV Recall	IV Recall
CharBased	0.334	0.012	0.485
MaxMatch	0.828	0.012	0.951

MT Performance

Segmenter	MT03 (dev)	MT05 (test)
CharBased	30.81	29.36
MaxMatch	31.95	30.73

Table 2.5: Segmentation and MT performance of the CharBased segmenter versus the MaxMatch segmenter.

The MaxMatch segmenter is a simple and common baseline for the Chinese word segmentation task, and is actually used in many real applications due its efficiency, easy implementation and easy integration with different lexicons.

The segmentation performance of MaxMatch is not very satisfying because it cannot generalize to capture words it has never seen before. However, having a basic segmenter like MaxMatch still gives the phrase-based MT system a win over the character-based segmentation (treating each Chinese character as a word). We will refer to the character-based segmentation as CharBased.

We first evaluate the two segmenters on the SIGHAN 2006 UPUC data. In Table 2.5, we can see that on the Chinese word segmentation task, having MaxMatch is obviously better than not trying to identify Chinese words at all (CharBased). We can see that all the improvement of MaxMatch over CharBased is on the recall rate of in-vocabulary words, which makes sense because MaxMatch does not attempt to recognize words that are not in the lexicon. As for MT performance, in Table 2.5 we see that having a segmenter, even as simple as MaxMatch, can help a phrase-based MT system by about 1.37 BLEU points on all 1082 sentences of the test data (MT05). Also, we tested the performance of both CharBased and MaxMatch on 828 sentences of MT05 where all elements are in vocabulary.[4] MaxMatch achieved 32.09 BLEU and CharBased achieved 30.28 BLEU, which shows that on the sentences where all elements are in vocabulary, there MaxMatch is still significantly better than CharBased. Therefore, Hypothesis 1 is refuted.

Analysis We hypothesized in Hypothesis 1 that the phrase table in a phrase-based MT

[4]Except for dates and numbers.

system should be able to capture the meaning of non-compositional words by building "phrases" on top of character sequences. Based on the experimental result in Table 2.5, we see that using character-based segmentation (CharBased) actually performs reasonably well, which indicates that the phrase table does capture the meaning of character sequences to a certain extent. However, the results also show that there is still some benefit in having word segmentation for MT. We analyzed the decoded output of both systems (CharBased and MaxMatch) on the development set (MT03). We found that the advantage of MaxMatch over CharBased is two-fold:

1. Lexical: MaxMatch enhances the ability to disambiguate the case when a character has very different meanings in different contexts.

2. Reordering: It is easier to move one unit around than having to move two consecutive units at the same time. Having words as the basic units helps the reordering model.

For the first advantage, one example is the character "智", which can both mean "intelligence", or an abbreviation for Chile (智利). Looking at this particular character "智", we provide one example to compare between CharBased and MaxMatch in Table 2.6. In the example, the word 失智症 (dementia) is unknown for both segmenters. However, MaxMatch gave a better translation of the character 智. The issue here is not that the "智"→"intelligence" entry never appears in the phrase table of CharBased. The real issue is, when 智 means Chile, it is usually followed by the character 利. So by grouping them together, MaxMatch avoided falsely increasing the probability of translating the stand-alone 智 into Chile. Based on our analysis, this ambiguity occurs the most when the character-based system is dealing with a rare or unseen character sequence in the training data, and also occurs more often when dealing with transliterations. The reason is that characters composing a transliterated foreign named entity usually doesn't preserve their meanings; they are just used to compose a Chinese word that sounds similar to the original word – much more like using a character segmentation of English words. Another example of this kind is "阿耳滋海默氏症" (Alzheimer's disease). The MT system using CharBased segmentation tends to translate some characters individually and drop others; while the system using MaxMatch segmentation is more likely to translate it right.

Reference translation: scientists complete sequencing of the chromosome linked to early dementia
CharBased segmented input: 科␣学␣家␣为␣攸␣关␣初␣期␣失␣智␣症␣的␣染␣色␣体␣完␣成␣定␣序
MaxMatch segmented input: 科学家␣为␣攸关␣初期␣失␣智␣症␣的␣染色␣体␣完成␣定␣序
Translation with CharBased segmentation: scientists at the beginning of the stake of chile lost the genome sequence completed
Translation with MaxMatch segmentation: scientists at stake for the early loss of intellectual syndrome chromosome completed sequencing

Table 2.6: An example showing that character-based segmentation provides a weaker ability to distinguish characters with multiple unrelated meanings.

The second advantage of having a segmenter like the lexicon-based MaxMatch is that it helps the reordering model. Results in Table 2.5 are with the linear distortion limit defaulted to 6. Since words in CharBased are inherently shorter than MaxMatch, having the same distortion limit means CharBased is limited to a smaller context than MaxMatch. To make a fairer comparison, we set the linear distortion limit in Moses to unlimited, removed the lexicalized reordering model, and retested both systems. With this setting, MaxMatch is 0.46 BLEU point better than CharBased (29.62 to 29.16) on MT03. This result suggests that having word segmentation does affect how the reordering model works in a phrase-based system.

Hypothesis 2. *Better Segmentation Performance Should Lead to Better MT Performance*

Observation We have shown in Hypothesis 1 that it is helpful to segment Chinese texts into words first. In order to decide which segmenter to use, the most intuitive thing to do is to find one that gives a high F-score on segmentation. Our experiments show that higher F-score does not necessarily lead to higher BLEU score. In order to contrast with the simple maximum matching lexicon-based model (MaxMatch), we built another segmenter with a CRF model. It is commonly agreed that with a CRF model, the segmenter can achieve better F-score than the MaxMatch segmenter. We want to show that even though

Segmentation Performance

Segmenter	F-score	OOV Recall	IV Recall
CRF-basic	0.877	0.502	0.926
MaxMatch	0.828	0.012	0.951
CRF-Lex	0.940	0.729	0.970

MT Performance

Segmenter	MT03 (dev)	MT05 (test)
CRF-basic	33.01	30.35
MaxMatch	31.95	30.73
CRF-Lex	32.70	30.95

Table 2.7: Segmentation and MT performance of the feature-based CRF-basic segmenter versus the lexicon-based MaxMatch segmenter

a segmenter has higher F-score on the segmentation evaluation, it can still be worse on a higher level task like MT. Therefore it is important to understand what the important characteristics are for an application instead of blindly trusting segmentation F-score.

We trained a CRF model with a set of basic features: character identity features of the current character, previous character and next character, and the conjunction of previous and current characters in the zero-order templates. We will refer to this segmenter as CRF-basic.

We evaluate both the feature-based segmenter CRF-basic and the lexicon-based MaxMatch segmenter on their segmentation F-score. Table 2.7 shows that CRF-basic outperforms MaxMatch by 5.9% relative F-score. Comparing the OOV recall rate and the IV recall rate, the reason is that CRF-basic wins a lot on the OOV recall rate. We see that a feature-based segmenter like CRF-basic clearly has stronger ability to recognize unseen words. On MT performance, however, CRF-basic is 0.38 BLEU points worse than MaxMatch on the test set. The reason may be that CRF-basic is overgenerating OOVs that are not actually OOVs. Since CRF-basic is using over simplified features, we introduce another CRF segmenter CRF-Lex that uses state-of-the-art features, including the ones in Section 2.2.2, a few improved features, and lexicon-based features described in Section 2.3.5. For further discussion, in Section 2.3.3, we will look at how the MT training and test data are segmented by each segmenter, and provide statistics and analysis for why certain segmenters are better than others.

Segmenter	#MT Training Lexicon Size	#MT Test Lexicon Size
CRF-basic	583,147	5443
MaxMatch	39,040	5083
CRF-Lex	411,406	5164
	MT Test Lexicon OOV rate	Conditional Entropy
CRF-basic	7.40%	0.2306
MaxMatch	0.49%	0.1788
CRF-Lex	4.88%	0.1010

Table 2.8: MT Lexicon Statistics and Conditional Entropy of Segmentation Variations of three segmenters

2.3.3 Consistency Analysis of Different Segmenters

In Section 2.3.2 we have refuted two hypotheses. Now we know that:

1. Phrase table construction does not fully capture what a word segmenter can do. Thus it is useful to have word segmentation for MT.

2. A higher F-score segmenter does not necessarily outperform on the MT task.

The table also shows another CRF segmenter, CRF-Lex, that includes lexicon-based features by using external lexicons. More details of CRF-Lex will be described in Section 2.3.5. To understand what factors other than segmentation F-score can affect MT performance, we now compare the three segmenters: CRF-Lex is a feature-based segmenter with better features and also includes lexicon-based features; CRF-basic is a feature-based segmenter with only very basic features, but still outperforming MaxMatch; MaxMatch is a lexicon-based maximum matching segmenter. We compare the segmentation and MT performance of these three in Table 2.7. In this table, we see that the ranking of segmentation F-score is:

$$\text{CRF-Lex} > \text{CRF-basic} > \text{MaxMatch}.$$

And now we know that the better segmentation F-score does not always lead to better MT BLEU score, because of in terms of MT performance,

$$\text{CRF-Lex} > \text{MaxMatch} > \text{CRF-basic}.$$

To explain this phenomenon, in Table 2.8, we list some statistics of each segmenter to help understand what are the desired characteristics of a segmenter for MT. First we look at the lexicon size of the MT training and test data. That is, the lexicon size is the number of distinct word tokens that result when we segment the MT training data with the given segmenter. While segmenting the MT data, CRF-basic generates an MT training lexicon size of 583K unique word tokens, and MaxMatch has a much smaller lexicon size of 39K. CRF-Lex performs best on MT, but the MT training lexicon size and test lexicon OOV rate is still high compared to MaxMatch. Examining only the MT training and test lexicon size still does not fully explain why CRF-Lex outperforms MaxMatch. MaxMatch generates a smaller MT lexicon and lower OOV rate, but for MT it was not better than CRF-Lex, which has a bigger lexicon and higher OOV rate. In order to understand why MaxMatch performs worse on MT than CRF-Lex but better than CRF-basic, we use conditional entropy of segmentation variations to measure consistency. This is motivated by the following new hypothesis.

Hypothesis 3. *For MT, word segmentation consistency is important to MT system performance.*

Observation We use the gold segmentation of the SIGHAN test data as a guideline. To define the conditional entropy of segmentation variations more clearly, here is a more concrete explanation. For every word type w_i, we collect all the different pattern variations v_{ij} in the segmentation we want to examine. For example, for a word "*ABC*" in the gold segmentation, we look at how it is segmented with a segmenter. There are many possibilities. If we use c_x and c_y to indicate other Chinese characters and ␣ to indicate white spaces, "c_x␣ABC␣c_y" is the correct segmentation, because the three characters are properly segmented from both sides, and they are concatenated with each other. It can also be segmented as "c_x␣A␣BC␣c_y", which means although the boundary is correct, the first character is separated from the other two. Or, it can be segmented as "c_xA␣BCc_y", which means the first character was actually part of the previous word, while *BC* are the beginning of the next word. Every time a particular word type w_i appears in the text, we consider a segmenter more consistent if it can segment w_i in the same way every time, but it does not necessarily

Segmenter	c_x␣人民␣c_y	c_x人民␣c_y	c_x␣人民c_y	c_x人␣民␣c_y	entropy
CRF-basic	159	1	17	0	0.506
MaxMatch	110	0	0	67	0.957
CRF-Lex	117	0	0	0	0

Table 2.9: Different variations of segmentation pattern and corresponding entropy of each segmenter for " 人民".

have to be the same as the gold standard segmentation. For example, if "*ABC*" is a Chinese person name which appears 100 times in the gold standard data, and one segmenter segments it as c_x␣A␣BC␣c_y 100 times, then this segmenter is still considered to be very consistent, even if it does not exactly match the gold standard segmentation. Using this intuition, the conditional entropy of segmentation variations $H(V|W)$ is defined as follows:

$$\begin{aligned} H(V|W) &= -\sum_{w_i} P(w_i) \sum_{v_{ij}} P(v_{ij}|w_i) \log P(v_{ij}|w_i) \\ &= -\sum_{w_i} \sum_{v_{ij}} P(v_{ij}, w_i) \log P(v_{ij}|w_i) \end{aligned}$$

To illustrate how the conditional entropy of segmentation variations can capture consistency, we give an example of different segmentation variations of the word "人民" (people). In the SIGHAN test data, "人民" occurs 177 times. The segmentation variations and frequencies of different segmenters are listed in Table 2.9. In this example, because the first character "人" is also likely to be a suffix, when doing left-to-right MaxMatch, it can be frequently confused as a suffix of the previous word. In fact, for the MaxMatch segmenter, 67 out of 177 times the word was wrongly segmented as c_x人␣民␣c_y.

Now we can look at the overall conditional entropy $H(V|W)$ to compare the consistency of each segmenter. In Table 2.8, we can see that even though MaxMatch has a much smaller MT lexicon size than CRF-Lex, when we examine the consistency of how MaxMatch segments in context, we find the conditional entropy is much higher than CRF-Lex. We can also see that CRF-basic has a higher conditional entropy than the other two. The conditional entropy $H(V|W)$ shows how consistent each segmenter is, and it correlates with the MT performance in Table 2.8.

Now we know that consistency is important for MT performance. A stronger hypothesis is to say better consistency always indicates better MT performance:

Hypothesis 4. *Better segmentation consistency always leads to better MT performance.*

Observation This hypothesis can be easily refuted by a counter example. For example, a character-based segmentation will always have the best consistency possible, since every word *ABC* will just have one pattern: c_x␣A␣B␣C␣c_y. But from Section 2.3.2 we see that CharBased performs worse than both MaxMatch and CRF-basic on MT, because having word segmentation can help the granularity of the Chinese lexicon match that of the English lexicon. So, consistency is only one of the competing factors of how good a segmentation is for MT performance.

In conclusion, for MT performance, it is helpful to have consistent segmentation, while still having a word segmentation matching the granularity of the segmented Chinese lexicon and the English lexicon.

2.3.4 Optimal Average Token Length for MT

In the previous sections, we have analyzed different segmentation techniques with radically different characteristics. On the one hand, a character-level segmenter can dramatically reduce test lexicon OOV rate, and almost guarantees all input tokens can be translated. On the other hand, some word-based segmenters do a better job at identifying minimal units of meaning (e.g., "天花" for smallpox), and yield translation quality superior to character-based systems.

We have shown earlier that word-level segmentation greatly outperforms character based segmentation in MT evaluations. Since the word segmentation standard under consideration (Chinese Treebank (Xue et al., 2005)) was neither specifically designed nor optimized for MT, it seems reasonable to investigate whether any segmentation granularity in continuum between character-level and CTB-style segmentation is more effective for MT. In this section, we present a technique for directly optimizing a segmentation property—characters per token average—for translation quality, which yields significant improvements in MT performance.

λ_0	-1	0	1	2	4	8	32
len	1.64	1.62	1.61	1.59	1.55	1.37	1

Table 2.10: Effect of the bias parameter λ_0 on the average number of character per token on MT data.

In order to calibrate the average word length produced by our CRF segmenter—i.e., to adjust the rate of word boundary predictions ($y_t = +1$), we apply a relatively simple technique (Minkov et al., 2006) originally devised for adjusting the precision/recall trade off of any sequential classifier. Specifically, the weight vector $\mathbf{w}$ and feature vector of a trained linear sequence classifier are augmented at test time to include new class-conditional feature functions to bias the classifier towards particular class labels. In our case, since we wish to increase the frequency of word boundaries, we add a feature function:

$$f_0(\mathbf{x}, y_{t-1}, y_t, t) = \begin{cases} 1 & \text{if } y_t = +1 \\ 0 & \text{otherwise} \end{cases}$$

Its weight λ_0 controls the extent to which the classifier will make positive predictions, with very large positive λ_0 values causing only positive predictions (i.e., character-based segmentation) and large negative values effectively disabling segmentation boundaries. Table 2.10 displays how changes of the bias parameter λ_0 affect segmentation granularity.[5] Since we are interested in analyzing the different regimes of MT performance between CTB segmentation and character-based, we performed a grid search in the range between $\lambda_0 = 0$ (maximum-likelihood estimate) and $\lambda_0 = 32$ (a value that is large enough to produce only positive predictions). For each λ_0 value, we ran an entire MT training and testing cycle, i.e., we re-segmented the entire training data, ran GIZA++, acquired phrasal translations that abide with this new segmentation, and ran MERT and evaluations on segmented data using the same λ_0 values.

Segmentation and MT results are displayed in Figure 2.3. First, we observe that adjusting the precision and recall trade-off by setting negative bias values ($\lambda_0 = -2$) slightly

[5]Note that character-per-token averages provided in the table consider each non-Chinese word (e.g., foreign names, numbers) as one character, since our segmentation post-processing prevents these tokens from being segmented.

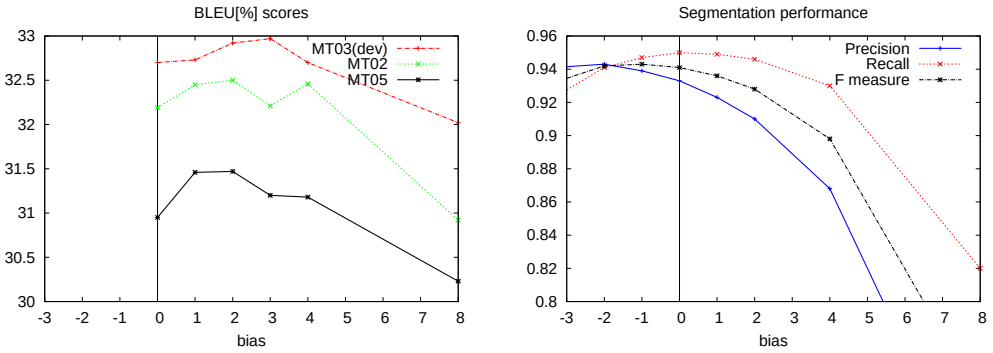


Figure 2.3: A bias towards more segment boundaries ($\lambda_0 > 0$) yields better MT performance and worse segmentation results.

improves segmentation performance. We also notice that raising λ_0 yields relatively consistent improvements in MT performance, yet causes segmentation performance (F-score) to be increasingly worse. While the latter finding is not particularly surprising, it further confirms that segmentation and MT evaluations can yield rather different outcomes. For our experiment testing this feature, we chose $\lambda_0 = 2$ on a second dev set (MT02). On the test set MT05, $\lambda_0 = 2$ yields 31.47 BLEU, which represents a quite large improvement compared to the unbiased segmenter (30.95 BLEU). Further reducing the average number of characters per token yields gradual drops of performance until it reaches character-level segmentation ($\lambda_0 \geq 32$, 29.36 BLEU).

Here are some examples of how setting $\lambda_0 = 2$ shortens the words in a way that can help MT.

- separating adjectives and pre-modifying adverbs:
  很大 (*very big*) → 很(*very*) 大(*big*)

- separating nouns and pre-modifying adjectives:
  高血压 (*high blood pressure*)
  → 高(*high*) 血压(*blood pressure*)

- separating compound nouns:
  内政部 (*Department of Internal Affairs*)
  → 内政(*Internal Affairs*) 部(*Department*).

Lexicon-based Features		Linguistic Features	
(1.1)	$L_{\text{Begin}}(C_n), n \in [-2,1]$	(2.1)	$C_n, n \in [-2,1]$
(1.2)	$L_{\text{Mid}}(C_n), n \in [-2,1]$	(2.2)	$C_{n-1}C_n, n \in [-1,1]$
(1.3)	$L_{\text{End}}(C_n), n \in [-2,1]$	(2.3)	$C_{n-2}C_n, n \in [1,2]$
(1.4)	$L_{\text{End}}(C_{-1}) + L_{\text{End}}(C_0)$ $+L_{\text{End}}(C_1)$	(2.4)	$\text{Single}(C_n), n \in [-2,1]$
		(2.5)	$\text{UnknownBigram}(C_{-1}C_0)$
(1.5)	$L_{\text{End}}(C_{-2}) + L_{\text{End}}(C_{-1})$ $+L_{\text{Begin}}(C_0) + L_{\text{Mid}}(C_0)$	(2.6)	$\text{ProductiveAffixes}(C_{-1}, C_0)$
		(2.7)	$\text{Reduplication}(C_{-1}, C_n), n \in [0,1]$
(1.6)	$L_{\text{End}}(C_{-2}) + L_{\text{End}}(C_{-1})$ $+L_{\text{Begin}}(C_{-1})$ $+L_{\text{Begin}}(C_0) + L_{\text{Mid}}(C_0)$		

Table 2.11: The lexicon-based and linguistic features for CRF-Lex

2.3.5 Improving Segmentation Consistency of a Feature-based Sequence Model for Segmentation

In Section 2.3.2 we showed that a statistical sequence model with rich features can generalize better than maximum matching segmenters. However, it also inconsistently overgenerates a big MT training lexicon and OOV words in MT test data, and thus causes a problem for MT. In this section, to improve a feature-based sequence model for MT, we propose 4 different approaches to deal with named entities, optimal length of word for MT and joint search for segmentation and MT decoding.

Making Use of External Lexicons

One way to improve the consistency of the CRF model is to make use of external lexicons (which are not part of the segmentation training data) to add lexicon-based features. All the features we use are summarized in Table 2.11. Our linguistic features are the ones already described in Section 2.2.2. Our lexicon-based features are adopted from (Shi and Wang, 2007), where $L_{\text{Begin}}(C_0)$, $L_{\text{Mid}}(C_0)$ and $L_{\text{End}}(C_0)$ represent the maximum length of words found in a lexicon that contain the current character as either the first, middle or last character, and we group any length equal or longer than 6 together. The linguistic features help in capturing words that were unseen to the segmenter; while the lexicon-based features constrain the segmenter with external knowledge of what sequences are likely to be words.

We built a CRF segmenter with all the features listed in Table 2.11 (CRF-Lex). The external lexicons we used for the lexicon-based features come from various sources including named entities collected from Wikipedia and the Chinese section of the UN website, named entities collected by Harbin Institute of Technology, the ADSO dictionary, EMM News Explorer, Online Chinese Tools, Online Dictionary from Peking University and HowNet. There are 423,224 distinct entries in all the external lexicons.

The MT lexicon consistency of CRF-Lex in Table 2.8 shows that the MT training lexicon size has been reduced by 29.5% and the MT test data OOV rate is reduced by 34.1%.

Joint training of Word Segmentation and Proper Noun Tagging

Named entities are an important source for OOV words, and in particular consider especially words which are bad to break into pieces (particularly for foreign names). Therefore, we use the proper noun (NR) part-of-speech tag information from CTB to extend the label set of our CRF model from 2 to 4 ($\{$beginning of a word, continuation of a word$\} \times \{$NR, not NR$\}$). This is similar to the "all-at-once, character-based" POS tagging in (Ng and Low, 2004), except that we are only tagging proper nouns. We call the 4-label extension CRF-Lex-NR. The segmentation and MT performance of CRF-Lex-NR is listed in Table 2.12. With the 4-label extension, the OOV recall rate improved by 3.29%; while the IV recall rate stays the same. Similar to (Ng and Low, 2004), we found the overall F-score only goes up a tiny bit, but we do find a significant OOV recall rate improvement.

On the MT performance, CRF-Lex-NR has a 0.32 BLEU gain on the test set MT05. In addition to the BLEU improvement, CRF-Lex-NR also provides extra information about proper nouns, which can be combined with postprocessing named entity translation modules to further improve MT performance.

2.4 Conclusion

Chinese word segmentation has been a complicated problem because of the intrinsic ambiguity of word definition, multiple existing segmentation standards, and the different needs

Segmentation Performance

Segmenter	F-score	OOV Recall	IV Recall
CRF-Lex-NR	0.943	0.753	0.970
CRF-Lex	0.940	0.729	0.970

MT Performance

Segmenter	MT03 (dev)	MT05 (test)
CRF-Lex-NR	32.96	31.27
CRF-Lex	32.70	30.95

Table 2.12: Segmentation and MT performance of CRF-Lex-NR versus CRF-Lex. This table shows the improvement of jointly training a Chinese word segmenter and a proper noun tagger both on segmentation and MT performance.

of different NLP applications. In this chapter, we introduced several morphologically inspired features in a conditional random fields framework. We showed that using morphological features prevents the segmenter from overfitting to a segmentation standard. This implementation was ranked highly in the SIGHAN 2005 Chinese word segmentation bakeoff contest.

In order to improved machine translation performance, we investigated which segmentation properties are important. First, we found that neither a character-based nor a standard word segmentation standard are optimal for MT, and show that an intermediate granularity is much more effective. Using an already competitive CRF segmentation model, we directly optimize segmentation granularity for translation quality, and obtain an improvement of 0.73 BLEU points on MT05 over our lexicon-based segmentation baseline. Second, we augment our CRF model with lexicon and proper noun features in order to improve segmentation consistency, which provides a 0.32 BLEU point improvement.

So far in this chapter we focused on producing one good segmentation that works best for MT. There is also value in considering segmentations of multiple granularities at decoding time. This will prevent segmentation error from propagating and can also help translations for compound words. One demonstration of this is recent work that uses segmentation lattices at decoding time to improve the overall MT quality (Dyer et al., 2008; Dyer, 2009).

Also, investigating different probabilistic models could help on the problem of segmentation for MT. For example, Andrew (2006) proposed a hybrid model of CRFs and

semi-Markov CRFs that outperformed either individually on the Chinese word segmentation task. This shows the potential of more sophisticated models to help segmentation for MT.

Chapter 3

Error Analysis of Chinese-English MT Systems

In this chapter, we analyze the output of three state-of-the-art machine translation systems. The systems we look at are the three teams that participated in the 2007 go/no go (GnG) test of the GALE phase 2 program. The GALE (Global Autonomous Language Exploitation) program evaluates machine translation technology, and focuses on the source languages Arabic and Chinese to the target language English. The input for translation can be either audio or text, and the output is text. In this chapter we only analyze the Chinese-to-English translation part with text input.

The three teams are Agile, Nightingale, and Rosetta. Each team is composed of several sites, including universities and company research labs. Stanford is part of the Rosetta team. All teams are allowed to use a shared set of training data for building the components in their MT systems. From a bird's eye view, the three teams all follow a similar pipeline – several independent MT outputs are generated by the groups within the teams, and then systems are combined to get a best candidate from all the possible candidates. If we look in more detail, many different types of MT systems (e.g., phrase-based systems, hierarchical phrase-based systems, rule-based systems, syntax-based systems, etc) are involved. The system combination approaches are also different among teams.

In Section 3.1, we first introduce the three systems. The system descriptions are from the GALE P2.5 system descriptions of each of the three teams. Understanding what are the

components in each team can help understand the MT outputs and why certain mistakes are common. Detailed analysis of the three systems is in Section 3.2.

3.1 GALE system descriptions

3.1.1 Agile

The Agile team used 9,865,928 segments, including 246,936,935 English tokens for training. Each site can decide to use a subset of the available training data. The Chinese text was tokenized and distributed to all sites by BBN. Each individual site returned a tokenized lower-cased N-best list from their systems. The outputs were combined at BBN using the approach proposed by Rosti et al. (2007), which is a confusion network based method for combining outputs from multiple MT systems.

The Agile team has 6 individual Chinese-English MT systems that they used in the system combination.

1. LW (Language Weaver) phrase-based MT system

2. ISI (Information Sciences Institute) hierarchical MT system (Chiang, 2005)

3. ISI/LW syntax-based MT system (Galley, Hopkins, Knight, and Marcu, Galley et al.; Marcu et al., 2006): the first three systems were run by ISI. All of the ISI systems used LEAF (Fraser and Marcu, 2007) word alignment instead of GIZA++.

4. Moses system (Koehn et al., 2007) from University of Edinburgh: for preprocessing, they also applied the reordering method introduced by Wang et al. (2007) to preprocess the training and test data.

5. Cambridge University system: the Cambridge University system is a weighted finite state transducer implementation of phrase-based translation. They also used a different word alignment tool – MTTK (Deng and Byrne, 2005).

6. BBN's hierarchical MT system HierDec (Shen et al., 2008): it is a hierarchical system similar to (Chiang, 2005), but augmented with English dependency treelets.

3.1.2 Nightingale

The Nightingale team has 7 different MT systems. The other six systems are:

1. NRC system: NRC used two word alignment algorithms: symmetrized HMM-based word alignment and symmetrized IBM2 word alignment, and extracted two phrase tables per corpus. They decoded with beam search using the cube pruning algorithm introduced in (Huang and Chiang, 2007).

2. NRC-Systran hybrid system: first, Systran's rule-based system input the Chinese source and outputs an initial "Systran English" translation, then the NRC statistical system tries to *translate* the "Systran English" to English. Thus, the NRC component of this system is trained on parallel corpora consisting of "Systran English" (translations of Chinese sentence) aligned with the English references. More details of the setup and the Systran component can be found in (Dugast et al., 2007).

3. RWTH phrase-based translation system: it used standard GIZA++ alignment, and decoded with a phrase-based dynamic programming beam-search decoder. One specialized component is a maximum entropy model to predict and preprocess Chinese verb tense (present, past, future, infinitive). The prediction model uses features of the source sentence words and syntactic information. The model is trained on word aligned data, where the "correct" tense is extracted from a parse of the English sentence.

4. RWTH chunk based translation system: the Chinese sentences were parsed and chunks were extracted from the parses. Then reordering rules are extracted from the data as well. The system was introduced by Zhang et al. (2007).

5. SRI hierarchical phrases translation system: the hierarchical phrases extraction followed (Chiang, 2005), and the decoding is a CKY-like bottom up parsing similar to (Chiang, 2007).

6. HKUST dynamic phrase sense disambiguation based translation: the HKUST system augmented the RWTH phrase-based system with their phrase sense disambiguation approach introduced in (Carpuat and Wu, 2007).

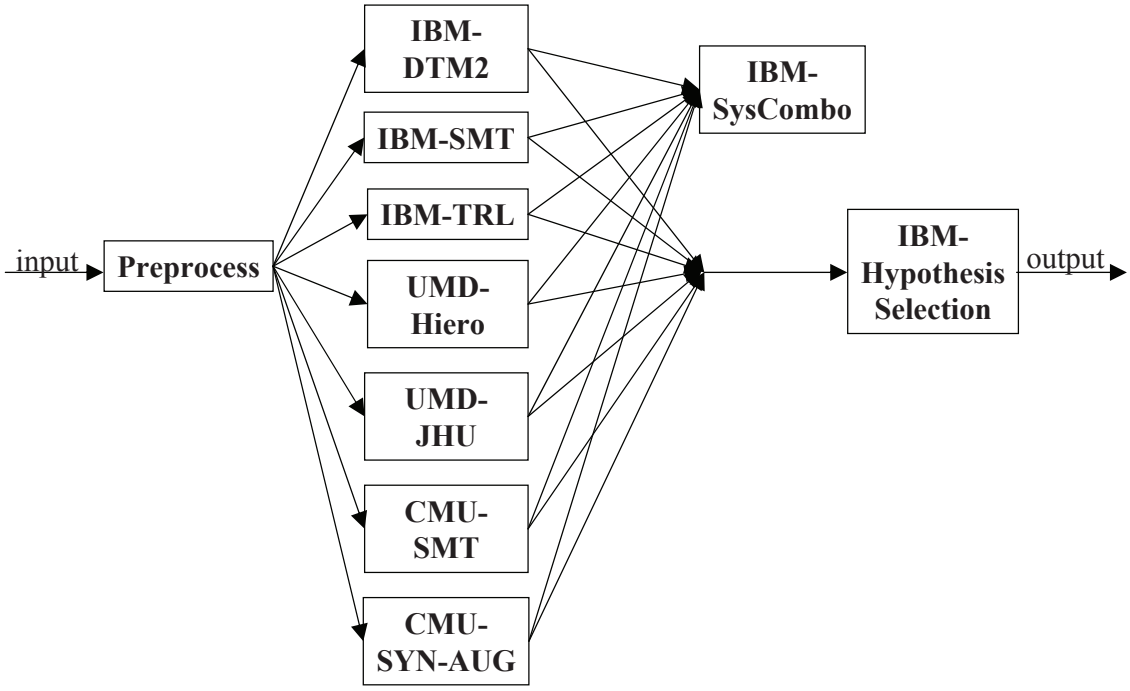


Figure 3.1: Workflow of the Rosetta MT systems.

7. The last MT system was a serial system combination of a rule-based and a statistical MT system. We don't have further information on this system.

The system combination is the confusion-network-based approach in (Matusov et al., 2006).

3.1.3 Rosetta

The training data processed and distributed team-wide in Rosetta contains 8,838,650 segments, 256,953,151 English tokens, and 226,001,339 Chinese tokens. The Rosetta team-wide preprocessing includes Stanford segmenter described in Chapter 2 and other IBM normalization components. Stanford didn't contribute an MT system in GALE phase 2, but began doing so in phase 3.

The Rosetta team has 7 individual MT systems. As illustrated in Figure 3.1, they were first combined with an IBM system combination module, and then the 7 system outputs as well as the combined output were sent to the IBM hypothesis selection module to generate

the final output. They first extract bilingual phrases, word alignments within phrases, and decoding path information from each system output. They also get the phrase table with IBM model 1 scores and decoding path cost of a baseline decoder. Based on this information, they re-decode the test set and generate the "IBM-SysCombo" output. In the second step of hypothesis selection, they select the best translation among multiple hypotheses (including IBM-SysCombo) using difference features, so systems not combined in the first step still have the opportunity to be selected in step 2. A more detailed description can be found in (Huang and Papineni, 2007).

The 7 individual MT systems are:

1. IBM-DTM2: For phrase extraction, simple blocks of style 1-M are extracted from alignments. Additionally non-compositional blocks are extracted only when the simple extraction fails yielding a very small number of additional blocks. Translation models used include IBM Model-1 scores in each direction, the unigram phrase probability, and the MaxEnt model described in "Direct Translation Model 2" (Ittycheriah and Roukos, 2007).

2. IBM-SMT: For phrase extraction, only contiguously aligned phrases (on both source and target side) are extracted. Exceptions are function words on both sides which are allowed to be unaligned but will still be included in the blocks (Tillmann and Xia, 2003). The decoder is a cardinality synchronous, multi-stack, multi-beam decoder, which was proposed in (Al-Onaizan and Papineni, 2006).

3. IBM-TRL: Phrases are extracted according to the projection and extension algorithms described in (Tillmann, 2003). Then the phrases are expanded to cover target words with no alignment links, as described in (Lee and Roukos, 2004). The translation models and scoring functions used in decoding are described in (Lee et al., 2006).

4. UMD-JHU: The system uses a hierarchical phrase-based translation model (Chiang, 2005), and decodes using a CKY parser and a beam search together with a postprocessor for mapping foreign side derivations to English derivations (Chiang, 2007). It also includes specialized components for Chinese abbreviation translation, named

	Rosetta	Agile	Nightingale
avg #toks per sent	28.59	29.30	34.02
avg #non-punct per sent	24.67	25.09	29.55
avg #non-function per sent	13.15	13.54	15.46

Table 3.1: Translation lengths of Rosetta, Agile, and Nightingale.

entities and number translations, and two-stage LM reranking.

5. UMD-Hiero: Also a hierarchical phrase-based system (Chiang, 2005, 2007).

6. CMU phrase-based SMT system (CMU-SMT): A log-linear model with about 13 features was used for phrase-pair (or block) extraction. They built a HM-BiTAM translation model using part of the training data. The STTK decoder then loads document-specific rescored phrase tables to decode the unseen test documents.

7. Syntax-Augmented SMT System (CMU-SYN-AUG): Decoding is done by the CKY algorithm extended to handle rules with multiple non-terminals.

3.2 Analysis

We conducted analysis on some sentences of the Chinese text portion of the 2007 GnG test set. We analyzed the unsequestered 24 sentences out of the first 50 contiguous sentences of a split of the data that was designated for error analysis by the IBM Rosetta consortium. We will present the analysis here.

Table 3.1 has the statistics on the average translation length of each system. We can see that overall Nightingale tends to generate longer translations than the other two systems, and Rosetta is slightly shorter than Agile.

We will list the sentences that we analyzed and provide concrete examples for what errors were generated by each system. In each example, we first present the source Chinese sentence and the reference English sentence. Also, we give the segmentation used in the Rosetta team so that we can see what errors segmentation might have caused. This segmentation was done by the Chinese word segmenter of Tseng et al. (2005) described in Section 2.2 of this thesis, and motivated the later work that appears in Section 2.3 on improving

segmentation consistency. In the analysis for each sentence, we discuss the performance of each system – whether they drop out important content words, capture complicated syntactic constructions, etc. In addition to that, in the analysis we point out interesting Chinese syntactic structures that are different from English syntax, and also discuss how they can cause difficulties in MT.

The number of a sentence is its linear order in the 2007 GnG set. Since the first sixteen sentences are sequestered, the first sentence we present has index 17.

- Sentence 17 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 2`)

  Source:

  戴蒙德在指出人们通常看到的白种人的优势–钢铁、技术、武器、集中统一的政府外，格外强调了一个鲜为人知的因素–细菌。

  Reference:

  In addition to pointing out the commonly seen advantages of the [Caucasians//white race]: steel, technologies, weapons, central and unified government, Desmond especially stressed one little-known factor – bacteria.

  Rosetta segmented source:

  戴蒙德 ␣在 ␣指出 ␣人们 ␣通常 ␣看到 ␣的 ␣白种人 ␣的 ␣优势 ␣- ␣- ␣钢铁 ␣, ␣技术 ␣, ␣武器 ␣, ␣集中 ␣统一 ␣的 ␣政府 ␣外 ␣, ␣格外 ␣强调 ␣了 ␣一 ␣个 ␣鲜为人知 ␣的 ␣因素 ␣- ␣- ␣细菌 ␣。

  Rosetta:

  The diamond in that people usually see the advantages of white people, steel, technology, weapons, and centralized government, particularly highlighted by an unknown factor, the bacteria.

  Nightingale:

  Dai Xianglong pointed out that people usually seen in the advantages of the White - iron and steel, technology, weapons, centralized government, particularly stressed the importance of a little-known factors - bacteria.

  Agile:

  In pointed out that people often see the advantage of the white people - iron and steel, technology, weapons, and unified government, a little-known factors - the bacteria is particularly.

Analysis:

(Note that the reference has an error – Desmond should be Diamond.) Rosetta mistranslated the name "戴蒙德" (Diamond) into "the diamond". Nightingale obviously has special treatment for names, but didn't get it either. Agile did worst on this sentence.

All three systems missed the double-embedded DE construction in the first clause ("commonly seen DE caucasian DE advantages", i.e. "commonly seen advantages of caucasians".) All three also missed the high-level structure of the whole sentence "In addition to pointing out X, Diamond emphasized Y". But Nightingale got the complex structure of that last clause ("particularly stressed...") spot on.

- Sentence 18 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 3`)

  Source:

  他说，在白种人征服新大陆 时，他们身上的细菌杀死的印第安人远远超过他们的武器。

  Reference:

  He said that when the [Caucasians//white race] conquered the new continent, the bacteria on their bodies killed far more Indians than their weapons.

  Rosetta segmented source:

  他 ␣说 ␣, ␣在 ␣白种人 ␣征服 ␣新 ␣大陆 ␣时 ␣, ␣他们 ␣身 ␣上 ␣的 ␣细菌 ␣杀死 ␣的 ␣印第 ␣安人 ␣远远 ␣超过 ␣他们 ␣的 ␣武器 ␣。

  Rosetta:

  White people, he said the conquest of new China, with Indians killed by far exceed their weapons.

  Nightingale:

  He said that in the white people to conquer new continent, they found the bacteria kill the Indians far exceed their weapons.

  Agile:

  He said that when the white people to conquer new continent, they found the bacteria killed are far more than their weapons.

  Analysis:

Rosetta dropped "细菌" (bacteria). Also Rosetta did an unnecessary reordering of "he said" and "when white people...". Rosetta mistranslated "大陆" (continent, mainland) into "China". Rosetta did worst on this sentence.

Also, there is a noun phrase with two DE constructions in the sentence:

他们	身	上	的	细菌	杀死	的	印第安人
they	body	on (localizer; SHANG)	(DE)	bacteria	kill	(DE)	Indians

(the Indians killed by the bacteria on their bodies)

If the translator tried to keep the Chinese noun phrase as an English noun phrase, the reference translation could have been: "..., the Indians killed by the bacteria on their bodies far exceeded the ones killed by their weapons." However, in the reference, instead of having 'killed by the bacteria' as a relative clause, the translator used the bacteria as the subject, and changed 'the Indians' to be the object, which might be a better translation, since later in the sentence 'the weapons' were also used to kill the Indians. This example shows that Chinese NPs with DE constructions can translate into complicated non-NP forms in English; it also shows that a Chinese constituent doesn't always map to an English constituent during translation.

- Sentence 19 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 4`)

  Source:

  1519年600名西班牙人在墨西哥登陆，去征服几百万人口的阿兹特克帝国，初次交锋他们损兵三分之二。

  Reference:

  In 1519, six hundred Spaniards landed on Mexico to conquer the Aztec Empire with a population of a few millions. They lost two thirds of their soldiers in the first clash.

  Rosetta segmented source:

  1519年 ␣600 ␣名 ␣西班牙人 ␣在 ␣墨西哥 ␣登陆 ␣, ␣去 ␣征服 ␣几百万 ␣人口 ␣的 ␣阿兹特克 ␣帝国 ␣, ␣初 ␣次 ␣交锋 ␣他们 ␣损兵 ␣三分之二 ␣。

  Rosetta:

  600 Spaniards and landing in Mexico, to conquer millions of the population of the Aztec empire, the first time, they have a 2/3.

  Nightingale:

1519 600 Spanish landing in Mexico, to conquer the millions of the population of the Aztec empire, they drove the initial damage soldiers, two thirds.

Agile:

In 1519 600 Spanish landed in Mexico, to conquer the Aztec empire of millions of people, the initial damage the two-thirds.

Analysis:

Rosetta dropped 1519 despite it being completely unambiguous (1519年). Rosetta also dropped the number 1,704 in seg 3 of AFP_CMN_20061211.0009-S1. It seems like Rosetta might have a particular problem with numbers. Rosetta also dropped "损兵" (loss of soldiers), which might be caused by a segmentation error. It might be better that "损兵" is segmented into two words "损" and "兵", where 损 is the verb meaning "to lose" and "兵" is the nominal object meaning "soldiers".

The NP with DE in this example, "几百万 (a few millions) 人口 (population) 的 (DE) 阿兹特克 (Aztec) 帝国 (empire)", should be reordered during translation. It should be "the Aztec Empire with a population of a few millions". Only Agile captured this reordering phenomenon, even though the translation wasn't perfect.

- Sentence 20 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 5`)

  Source:

  使他们取得决定性优势的是天花。

  Reference:

  What gave them the decisive advantage was smallpox.

  Rosetta segmented source:

  使 ␣他们 ␣取得 ␣决定性 ␣优势 ␣的 ␣是 ␣天 ␣花 ␣。

  Rosetta:

  smallpox to obtain a decisive advantage.

  Nightingale:

  To enable them to a decisive advantage of smallpox.

  Agile:

  That they achieved a decisive advantage is smallpox.

  Analysis:

Rosetta did an interesting reordering to move "smallpox" from the end of the sentence to the beginning. However, it doesn't quite complete the correct syntax of the sentence. The sentence could be translated as: "What gave them the decisive advantage was smallpox." Or, "(It was) Smallpox (that) gave them the decisive advantage."

- Sentence 21 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 6`)

  Source:

  1520年西班牙人传染给对方的天花病杀死了阿帝国的一半人，包括其皇帝。

  Reference:

  In 1520 the Spaniard infected the opponent with smallpox, killing half of the people in the Aztec Empire including its emperor.

  Rosetta segmented source:

  1520年 ␣西班牙人 ␣传染 ␣给 ␣对方 ␣的 ␣天 ␣花病 ␣杀死 ␣了 ␣阿帝国 ␣的 ␣一半人 ␣, ␣包括 ␣其 ␣皇帝 ␣。

  Rosetta:

  Spaniards transmitted to the other side a half, including its emperor.

  Nightingale:

  1520 Spain was transmitted to the other side of the smallpox disease killed his empire, half of the people, including the emperor.

  Agile:

  The Spanish transmitted to the other side's smallpox disease killed empire in half of the people, including the emperor.

  Analysis:

  Rosetta and Agile dropped the year (1520). Sadly, our segmenter mis-segments "天花病" (smallpox disease) into "天 / 花病"(sky* / flower disease*), so the Rosetta system dropped words again. Then, "阿帝国", which is an abbreviation of "阿兹特克帝国" (Aztec empire), probably stays untranslated and is dropped again. For the other 2 systems, it seems like their segmenters decided to separate it into "阿 / 帝国", so they at least get the "empire" part. The NP with DE construction in the source sentence "西班牙人传染给对方的天花病" (smallpox that the Spaniard infected the opponent with) was translated as a sentence instead of an NP. The reference could

also be "In 1520, smallpox that the Spaniard infected the opponent with killed half of the Aztec Empire including its emperor." This is an alternative translation that let the English translation of DE construction remain an NP. None of the systems translated this DE construction correctly.

- Sentence 22 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 7`)
  Source:
  1618年墨西哥的2000万人因传染病 减少到160万。
  Reference:
  In 1618 Mexico's twenty million people were reduced to 1.6 million due to the epidemic.
  Rosetta segmented source:
  1618年 ␣墨西哥 ␣的 ␣2000万 ␣人 ␣因 ␣传染 ␣病 ␣减少 ␣到 ␣160万 ␣。
  Rosetta:
  Mexico twenty million people were reduced to 1.6 million from communicable diseases.
  Nightingale:
  Due to enter Mexico's 20 million people of infectious diseases were reduced to 1.6 million.
  Agile:
  Mexico's 20 million people from communicable diseases reduced to 1.6 million.
  Analysis:
  All systems dropped the year "1618年". It is unclear why this happened in all three systems. Rosetta chose noun compound DE translation strategy here, where a possessive "'s" as in Nightingale's translation is actually more appropriate. Otherwise Rosetta did a pretty good job here; it captures what is the cause and what is the effect.

- Sentence 23 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 8`)
  Source:
  与此同时，印第安人缺乏一种细菌可以有效地打击对方。
  Reference:

In the meantime, the Indians lacked a kind of bacteria to effectively attack the opponent party.

Rosetta segmented source:

与 ␣此 ␣同时 ␣, ␣印第 ␣安人 ␣缺乏 ␣一 ␣种 ␣细菌 ␣可以 ␣有效 ␣地 ␣打击 ␣对方 ␣。

Rosetta:

In the meantime, the Indians lack of a type of bacteria can effectively crack down on the other side.

Nightingale:

At the same time, the Indians lack of a type of bacteria can be effective against each other.

Agile:

At the same time, the Indians lack a bacteria can be effective against each other.

Analysis:

Only Rosetta correctly translated "对方" into "the other side". The other 2 translate it into "each other". Other than that, none of them correctly made "可以有效地打击对方" into a relative clause "which can effectively crack down the other side". All three systems chose a present tense main verb "lack" even though past tense is appropriate here in the article context. This is an example of how MT systems, which usually translate sentence-by-sentence, do not handle tense selection in Chinese-English MT well. Some of the systems did mention they had modules for predicting Chinese verb tense, however it wasn't clear if discourse information was used.

- Sentence 24 (`DOC cmn-NG-31-111576-3460873-S1 S1 seg 9`)

  Source:

  戴蒙德没有将细菌归因于人种，而是认为它们与钢铁、枪炮一同直接或间接地起源于农业

  Reference:

  Desmond did not attribute bacteria to the human species, but believed that they, together with steel, guns and cannons, directly or indirectly originated from farming.

  Rosetta segmented source:

戴蒙德 ␣没有 ␣将 ␣细菌 ␣归因于 ␣人种 ␣, ␣而 ␣是 ␣认为 ␣它们 ␣与 ␣钢铁 ␣, ␣枪炮 ␣一同 ␣直接 ␣或 ␣间接 ␣地 ␣起源 ␣于 ␣农业

Rosetta:

Dimond did not be attributed to the advancement of the bacteria, is that they are the steel, and with directly or indirectly, to the origins of Agriculture

Nightingale:

Damon de bacteria will not attributed to the people, but of the race that they, together with steel guns directly or indirectly, originated in agriculture.

Agile:

Bacteria is not attributed to race, but that the iron and steel, guns were directly or indirectly originated in the agricultural industry

Analysis:

It's good that Rosetta translated the name of the author this time, but it didn't translate "人种" (race). It dropped "枪炮" (guns) as well. None of the systems got that the second verb "认为" (believes) should also have "戴蒙德" (Diamond) as its subject. In general, the output of all these systems is sufficiently poor that it is hard to know what is going on. Note that again, Desmond is incorrect for Diamond in the reference translation.

- Sentence 25 (`DOC cmn-NG-31-111868-3475012 S1 seg 1`)

  Source:

  感恩节一过，美国进入年末零售旺期。

  Reference:

  Following right after the Thanksgiving Day, the year-end shopping season kicked off in the US.

  Rosetta segmented source:

  感恩节 ␣一 ␣过 ␣, ␣美国 ␣进入 ␣年 ␣末 ␣零售 ␣旺期 ␣。

  Rosetta:

  A thanksgiving retail at the end of the year, the United States.

  Nightingale:

  A Thanksgiving Day, the US has entered a period of retail boom at the end of the

year.

Agile:

Thanksgiving, the United States entered the retail period at the end of the year.

Analysis:

None of the three systems captured the construction for "following right after A", where in Chinese the pattern is "A 一过". Rosetta dropped "进入" (enter), and "旺期" (peak season). This is a short sentence, and the segmentation seems fine. It might worth investigating why Rosetta did so badly on this short sentence.

- Sentence 26 (`DOC cmn-NG-31-111868-3475012 S1 seg 2`)

  Source:

  在各类卖场人头攒动之际，主要的购物拍卖网站也迎来了大量在线查询和购物的访客，这些网站的点击率、流量和销售也都因此大增。

  Reference:

  While various kinds of stores are [crowded with people//so crowded that one could only see the heads of people moving], the main shopping and auction websites have also ushered in a large number of visitors for online inquiries and shopping, with hit rates, traffic volume, and sales at these websites dramatically increased because of this.

  Rosetta segmented source:

  在 ␣各 ␣类 ␣卖场 ␣人头 ␣攒动 ␣之际 ␣, ␣主要 ␣的 ␣购物 ␣拍卖 ␣网站 ␣也 ␣迎来 ␣了 ␣大量 ␣在 ␣线 ␣查询 ␣和 ␣购物 ␣的 ␣访客 ␣, ␣这些 ␣网站 ␣的 ␣点击率 ␣, ␣流量 ␣和 ␣销售 ␣也 ␣都 ␣因此 ␣大增 ␣。

  Rosetta:

  All such stores of the teeming, the main shopping auction Web site also has ushered in a large number of visitors, users of these websites, traffic and sales.

  Nightingale:

  All kinds of stores in the teeming on the occasion, the main shopping online auction website also ushered in a large number of online shopping inquiries and visitors to the web sites of the click rate of flow and sales have increased.

  Agile:

The various kinds of tens of thousands of visitors, major shopping auction website also ushered in a large number of visitors and on-line shopping, the click rate, traffic and sales of these websites have multiplied.

Analysis:

None of the systems gets the first part of the Chinese sentence right. In the first clause of the Chinese sentence, we can see two different difficulties:

在	各	类	卖场	人头	攒动	之际
(preposition; ZAI)	various	kind	stores	human head	huddle together	while (localizer; ZHI-JI)

(while various kinds of stores are crowded with people)

First of all, the localizer phrase (LCP) "在(P) X 之际(LC)" should be translated as "while X"; none of the systems got this right. Second, there is a Chinese idiomatic expression here – "人头 攒动" literally describes the view when so many people gather together, you can only see their heads huddle together. It should be translated as "crowded". Interestingly, all three systems captured this meaning either by using the word "teeming" or "tens of thousands of visitors". But their translations are still far from understandable.

Rosetta did the worst. It also completely dropped the last part "也都因此大增'.

- Sentence 27 (`DOC cmn-NG-31-111868-3475012 S1 seg 3`)

  Source:

  全球最大的在线拍卖网站eBay.com以及其下属的货比三家类网站 Shopping.com 称，在感恩节过后的这个周五（今年是11月24日），其网站上最抢 手的货品是纪念版的大眼毛毛T.M.X.Elmo玩偶。

  Reference:

  EBay.com, the world's largest online auction website, and its subsidiary price comparison website Shopping.com said that this Friday (November 24 of this year) after Thanksgiving, the hottest merchandise on its website was the commemorative big-eye plush doll T.M.X. Elmo.

  Rosetta segmented source:

  全球 ␣最 ␣大 ␣的 ␣在 ␣线 ␣拍卖 ␣网站 ␣eBay ␣. ␣com ␣以及 ␣其 ␣下属 ␣的 ␣货比

三家类 ␣网站 ␣Shopping ␣. ␣com ␣称 ␣, ␣在 ␣感恩节 ␣过 ␣后 ␣的 ␣这 ␣个 ␣周五 ␣(␣今年 ␣是 ␣11月 ␣24日 ␣) ␣, ␣其 ␣网站 ␣上 ␣最 ␣抢手 ␣的 ␣货品 ␣是 ␣纪念版 ␣的 ␣大 ␣眼 ␣毛毛T.M.X.Elmo ␣玩偶 ␣。

Rosetta:

The world's largest online auction site eBay, as well as its said that in, in the Friday after Thanksgiving (November 24th on its web site, most of the editions of the plush doll.

Nightingale:

The world's largest online auction website and its subordinate eBay. Com comparison shopping Web site Shopping. Com category, in the wake of the Thanksgiving Day this Friday (November 24), and its web site this year is the most sought-after cinematographer of goods is a commemorative edition of the big eyes caterpillars T.m.x.elmo puppets.

Agile:

The world's largest online auction site Ebay.com and its three such websites Shopping.com said that after Thanksgiving this Friday (November 24 this year is), the goods of its Web site most is the version of the eyes t.m.x.elmo puppets.

Analysis:

Before the first comma, "...其下属的货比三家类网站Shopping.com称", Rosetta dropped the underscored part. The segmentation is: "下属 / 的 / 货比三家类 / 网站". It's not quite clear if it is correct or not to segment "货比三家类" all together. The first 4 characters "货比三家" is a Chinese idiom meaning "it's better to compare the price of 3 stores" and the last character means "type". In this sentence, "货比三家类" means "price-comparing". In terms of segmentation it might be better to segment it as "货比三家 / 类". Also, "Shopping.com" is an apposition of the noun phrase "下属的货比三家类网站". Rosetta completely dropped "Shopping.com"; Nightingale incorrectly used "subordinate" to modify eBay instead of Shopping.com; Agile mis-translated 货比三家 (price-comparing) and dropped 下属 (subordinate).

Rosetta had more problems of dropping words; it also dropped too much of the last part. The "T.M.X. Elmo" part might be dropped due to an segmentation error "毛毛T.M.X.Elmo".

- Sentence 28 (`DOC cmn-NG-31-111868-3475012 S1 seg 4`)

  Source:

  eBay.com称刚过去的这个”黑色周五”其网站上共拍卖出2537个大眼毛毛玩偶，均价70美元左右。

  Reference:

  EBay.com said that this past ”Black Friday” a total of 2,537 big-eye plush Elmo dolls were auctioned at an average price of around 70 US dollars at its website.

  Rosetta segmented source:

  eBay ␣. ␣com ␣称 ␣刚 ␣过去 ␣的 ␣这 ␣个 ␣” ␣黑色 ␣周五 ␣” ␣其 ␣网站 ␣上 ␣共 ␣拍 ␣卖出 ␣2537 ␣个 ␣大 ␣眼 ␣毛毛 ␣玩偶 ␣, ␣均价 ␣70 ␣美元 ␣左右 ␣。

  Rosetta:

  eBay. That just this past ”Black Friday” on its web site, sold 2537 in the plush toys, the average is about $70.

  Nightingale:

  EBay. Com claimed that the ” black Friday ” on its web site a total of 2537 big eyes plush toys, the average price of about US $70%.

  Agile:

  Ebay.com just past the ”Black Friday” on its Web site out a total of a big eyes fuzzy doll, the average price of around US $70.

  Analysis:

  This Chinese sentence can be quite difficult to translate because of the lack of punctuations. If there were a few more commas in the original Chinese sentence, it might be easier to figure out the sentence structure. First of all, there could be a comma after the verb 称 (claim; say). Also, if there were a comma right after “黑色周五” (Black Friday), it would be more clear that 刚过去的这个”黑色周五” is a temporal noun phrase *this past “Black Friday”*. In this example, Rosetta didn’t translate the first verb 称 (claim; say). Neither did Agile.

- Sentence 29 (`DOC cmn-NG-31-111868-3475012 S1 seg 5`)

  Source:

  同样的玩具在沃尔玛不到40美元，但很难有货，而在Shopping.com网站上其

最高成交价格是150美元

Reference:

The same toy is sold for less than 40 US dollars at Wal-Mart, but it is seldom in stock. Yet at the shopping.com website, the highest price it was sold for was 150 US dollars.

Rosetta segmented source:

同样 ␣的 ␣玩具 ␣在 ␣沃尔 ␣玛不到 ␣40 ␣美元 ␣, ␣但 ␣很 ␣难 ␣有货 ␣, ␣而 ␣在 ␣Shopping ␣. ␣com ␣网站 ␣上 ␣其 ␣最高 ␣成交 ␣价格 ␣是 ␣150 ␣美元

Rosetta:

The same toys at Wal-less than 40 US dollars, but it is very difficult, but in comparison, at $Its highest closing price of $150

Nightingale:

The same toys at Wal-Mart is less than US $40, but it is very difficult to have the goods, while in Shopping. Com web site its highest closing price of $150

Agile:

The same in Wal-Mart at less than US $40, but it is very difficult to goods, Shopping.com Web site its highest sale price is 150

Analysis:

The wrong translation "Wal-less" of Rosetta is due to wrong segmentation. In the first segment of the Chinese sentence, "沃尔玛" is the transliteration of Walmart, and "不到" means *less than*. But it was incorrectly segmented as "沃尔 / 玛不到". Segmenting "有货"(have the goods) together doesn't seem a smart decision either, but might have worked if it were translated as the idiom "in stock". No system captures this part of the sentence correctly. Again, Rosetta dropped the obvious "shopping.com" part. Overall this sentence is a combination of wrong segmentation and more word dropping.

- Sentence 33 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 2`)

  Source:

  我的童年很孤独，因为家里只有我一个孩子，周围的玩伴也大多搬到城里去了。

Reference:

I had a very lonely childhood because I was the only child in the family, and most of my playmates in the neighborhood had moved to the cities.

Rosetta segmented source:

我 ␣的 ␣童年 ␣很 ␣孤独 ␣, ␣因为 ␣家 ␣里 ␣只 ␣有 ␣我 ␣一 ␣个 ␣孩子 ␣, ␣周围 ␣的 ␣玩伴 ␣也 ␣大多 ␣搬到 ␣城 ␣里 ␣去 ␣了 ␣。

Rosetta:

my childhood was very lonely, because I have a child in the family, most of the friends around the city.

Nightingale:

My childhood was very lonely, because I am the only one child, around the playmates also have moved to go to the city.

Agile:

Most of my childhood playmates around very lonely, because I was the only one at home, also moved to the city.

Analysis:

Agile did a funny thing where they moved the "playmates" part into the first "lonely childhood" part. This may well show long distance reordering in a syntactic MT system going haywire. Rosetta seems to be confused about the meaning of "只有". This is due to the ambiguity of the word "有", which can mean "there exists" or "have (possession)". In this case it should be "there's only". Since "有" can often be translated into "have", Rosetta gave a rather funny translation: it should be "I was the only child", but Rosetta said "I have a child".

- Sentence 34 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 3`)

  Source:

  生活和家门前的那条小河一样，平静地流淌，没有一丝浪花。

  Reference:

  Life, like that small river in front of my home, flowed quietly without the slightest ripple.

  Rosetta segmented source:

生活 ␣和 ␣家门 ␣前 ␣的 ␣那 ␣条 ␣小河 ␣一样 ␣, ␣平静 ␣地 ␣流淌 ␣, ␣没有 ␣一 ␣丝 ␣浪花 ␣。

Rosetta:

life and in front of the river, flows calmly, without a trace.

Nightingale:

Life and in front of the door of the river, flows calmly, without a trace spray.

Agile:

Life and in front of the river, flows calmly, without a shred of the spray.

Analysis:

The Chinese "和" (3rd character in source) often can be translate into "and". However, in this case it's "A和B一样", which should be translated into "A is similar to B". All 3 systems use the "and" translation, which didn't get the meaning at all.

- Sentence 35 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 4`)

  Source:

  我经常一个人在河边看来往的行船，都是很小的渔船，一个篷下就是一个住家，我不知道他们从哪里来又到哪里去， 似乎从来没有看过同一条船经过。

  Reference:

  I often stayed on the riverbank by myself, watching the boats pass by. They were all very small fishing boats, and under each canopy was a household. I did not know where they came from or where they were going. It seemed that I never saw the same boat [pass by twice//pass by].

  Rosetta segmented source:

  我 ␣经常 ␣一 ␣个 ␣人 ␣在 ␣河 ␣边 ␣看 ␣来往 ␣的 ␣行船 ␣, ␣都 ␣是 ␣很 ␣小 ␣的 ␣渔船 ␣, ␣一 ␣个 ␣篷下 ␣就 ␣是 ␣一 ␣个 ␣住家 ␣, ␣我 ␣不 ␣知道 ␣他们 ␣从 ␣哪里 ␣来 ␣又 ␣到 ␣哪里 ␣去 ␣, ␣似乎 ␣从来 ␣没有 ␣看 ␣过同 ␣一 ␣条 ␣船 ␣经过 ␣。

  Rosetta:

  I often a person look at the sea from the river, they are very small fishing boats, a is a home, I don't know where they come from and to where it had never seen the same boat.

  Nightingale:

I often see a person in the river to the ship, is a very small fishing boats, under a canopy is a home, I do not know where they come from and where to go, it seems to have never seen in the same boat.

Agile:

I often to on the banks of the river, are very small fishing boats, under a canopy of a person is a home, I do not know where they come from and where to go, seems to have never seen the same boat.

Analysis:

This Chinese sentence is composed of several sentences when translated into English. The first part "我经常一个人在河边看来往的行船" is one sentence. One thing to notice is the pattern "一个人" means "by oneself" (in this case, "by myself"). It is a relatively common pattern, but maybe not in newswire. None of the systems translate this correctly. The second part of the Chinese sentence omitted the subject, which refers back to the boats in the first part. Considering these two pieces, Rosetta did a decent job, even though it translated "by myself" to "a person". But after that, Rosetta dropped "篷下" (under a canopy). This is probably due to the fact that the segmentation did not separate the noun "篷" and the preposition "下".

- Sentence 36 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 5`)

  Source:

  我母亲有时候问："华子，你去河边干嘛？"

  Reference:

  My mother sometimes asked: "Huazi, why do you go to the riverbank?"

  Rosetta segmented source:

  我 ␣母亲 ␣有 ␣时候 ␣问 ␣: ␣" ␣华子 ␣, ␣你 ␣去 ␣河 ␣边 ␣干嘛 ␣? ␣"

  Rosetta:

  My mother when asked: "What, you go to the river?"

  Nightingale:

  My mother asked: " Sometimes Hua Zi, you go to the river, why? "

  Agile:

  My mother sometimes asked: "Son, you go to the riverside, why?"

Analysis:

Rosetta mistranslated “有时候” to “when” because the segmentation was “有 / 时候”, where the first word means “exist” or “have” and the second word means “time” or “when”. If it was segmented as one word “有时候” (sometimes), it would be easier to translate it as *sometimes*.

The Chinese question ”你去河边干嘛” is a statement with a question marker at the end. Translating it directly will be “you go to the riverside do what”. Nightingale and Agile are both just translating it directly and put a “why” at the end, which is acceptable but not the most fluent English translation.

- Sentence 37 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 6`)

  Source:

  我说：”看船。”

  Reference:

  I said: ”To watch the boats.”

  Rosetta segmented source:

  我 ␣说 ␣: ␣” ␣看 ␣船 ␣。 ␣”

  Rosetta:

  I said: ”,” he said.

  Nightingale:

  I said: ” look at the boat. ”

  Agile:

  I said: ”Look at the boat.”

  Analysis:

  Rosetta is doing really bad in this case, which is hard to understand since the sentence is short and the segmentation is correct. One possible guess is the language model weight is too strong and not picking a translation with the right word “boat” in it. Although Nightingale and Agile translate the verb, they have problems choosing the right verb form: a “to” infinitive is required in English. They choose to use the bare verb, which has an imperative reading.

- Sentence 38 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 7`)

Source:
河的两头我都看不见，我开始想，这个世界应该是很大的，但它不属于我，我只是站在岸边看。
Reference:
I could not see either end of the river. I started to think that this world must be very large, but that it does not belong to me. I just stood on the riverbank watching.
Rosetta segmented source:
河 ␣的 ␣两 ␣头 ␣我 ␣都 ␣看 ␣不 ␣见 ␣, ␣我 ␣开始 ␣想 ␣, ␣这 ␣个 ␣世界 ␣应该 ␣是 ␣很 ␣大 ␣的 ␣, ␣但 ␣它 ␣不 ␣属于 ␣我 ␣, ␣我 ␣只是 ␣站 ␣在 ␣岸 ␣边看 ␣。
Rosetta:
River two in the first, I would not be able to see, I began to think that the world should be very big, but it does not belong to me, I just stood at the.
Nightingale:
I do not see at both ends of the river, I began to think that the world should be a very big, but it does not belong to me, I just stood on the shore watching.
Agile:
Both ends of the river I see, I began to think that the world should be very big, but it does not belong to me, I just stand on the dock.
Analysis:
The first part of the sentence is an OSV structure (topicalization). Only Nightingale got it right.

Rosetta dropped the last part because of wrong segmentation "岸 / 边看". (should be "岸边 (riverbank) / 看 (watch)") Other than those, this sentence seems to be easy.

- Sentence 39 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 8`)
Source:
八岁那年我们也搬家了，住到了小镇中心的一条街上。
Reference:
The year when I was eight, we also moved to live on a street in the center of a small town.
Rosetta segmented source:

八 ␣岁 ␣那 ␣年 ␣我们 ␣也 ␣搬家 ␣了 ␣, ␣住 ␣到 ␣了 ␣小 ␣镇 ␣中心 ␣的 ␣一 ␣条 ␣街 ␣上 ␣。

Rosetta:

Eight years old, we have also moved to live in a small town on a street in the centre.

Nightingale:

Eight years old, we have also moved to live in the town centre of a street.

Agile:

Eight-year-old that year we also, in a street in the town centre.

Analysis:

There's a zero in front of "八岁那年" (eight-year old that year). It should be "the year when I was eight years old". Nightingale mishandles the NP DE modification "小镇中心的一条街", which should be reordered as "a street in the town centre".

- Sentence 40 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 9`)

  Source:

  宝成伯依旧一个人孤单地住在河边，我还是会去看他，逢年过节父母会请他来吃饭。

  Reference:

  Uncle Baocheng still lived a lone life by the river by himself. I would still go visit him. On New Year's Day and other holidays, my parents would invite him over for meals.

  Rosetta segmented source:

  宝成 ␣伯 ␣依旧 ␣一 ␣个 ␣人 ␣孤单 ␣地住 ␣在 ␣河 ␣边 ␣, ␣我 ␣还 ␣是 ␣会 ␣去 ␣看 ␣他 ␣, ␣逢年 ␣过节 ␣父母 ␣会 ␣请 ␣他 ␣来 ␣吃饭 ␣。

  Rosetta:

  Pou Chen Bo as a person alone accommodated in the river, I will go to see him, holiday parents would ask him to dinner.

  Nightingale:

  The Baoji-Chengdu Hebron remains a lonely people to live in the river, I still go to see him on holidays, the parents would ask him to eat.

  Agile:

Baocheng, a single people live along the banks of the river, I still went to see him, parents and asked him to eat.

Analysis:

Rosetta did a good job translating the structure and meaning, but did not translate the first part very fluently. Nightingale wrongly put the time adverb clause to modify the first verb.

- Sentence 41 (`DOC cmn-NG-31-112514-3491242-S1 S1 seg 10`)

  Source:

  一晃，我十二岁了，进了当地一所名声不佳的中学。

  Reference:

  In a flash, I was twelve and attending a local middle school that had a bad reputation.

  Rosetta segmented source: 一晃 ␣, ␣我 ␣十二 ␣岁 ␣了 ␣, ␣进 ␣了 ␣当地 ␣一 ␣所 ␣名声 ␣不 ␣佳 ␣的 ␣中学 ␣。

  Rosetta:

  soon, I was 12 years old, fell into a local stigma secondary schools.

  Nightingale:

  In a flash, I am twelve years old, had a bad reputation of the local secondary school.

  Agile:

  In a flash, I was 12 years old, into the local a bad reputation secondary school.

  Analysis:

  In the Chinese sentence, there is an NP with DE construction: “一所名声不佳的中学”:

一	所	名声	不	佳	的	中学
one	(measure word)	reputation	not	good	(DE)	middle school

  (middle school that had a bad reputation)

  The NP needs to be reordered when translated into English. It should either be a prepositional phrase “with a bad reputation” or a relative clause “that had a bad reputation”. None of the systems got the reordering right. Even though Nightingale seemed to capture the prepositional phrase structure, they actually did not reorder correctly, and besides, they translated the verb “enter” as “had”, which totally lost

the meaning.

- Sentence 42 (DOC cmn-NG-31-112514-3491242-S1 S1 seg 11)

  Source:

  实际上我小学毕业考试 成绩很好，但因为父母没有门路，被最好的一中拒之门外，我有点消沉。

  Reference:

  Actually I graduated from elementary school with very good grades, but because my parents were not well connected, I was rejected by the best one, No.1 Middle School. I was somewhat despondent.

  Rosetta segmented source: 实际 ␣上 ␣我 ␣小学 ␣毕业 ␣考试 ␣成绩 ␣很 ␣好 ␣, ␣但 ␣因为 ␣父母 ␣没有 ␣门路 ␣, ␣被 ␣最 ␣好 ␣的 ␣一 ␣中 ␣拒 ␣之 ␣门 ␣外 ␣, ␣我 ␣有点 ␣消沉 ␣。

  Rosetta:

  In fact, I graduated from elementary school examination results are very good, but because the parents do, to the best in the door, I was a bit depressed.

  Nightingale:

  In fact, my primary school leaving examination results are very good, but because their parents do not have access, was one of the best in the door, I am a bit depressed.

  Agile:

  In fact, my primary leaving examination results are very good, but because their parents do not have connections, was the best in the door, I am a little bit depressed.

  Analysis:

  There are many zeros in the sentence. This is a case of topic-continuous zeros, where every verb after the second clause has a zero whose subject refers to “I”. For example, right before “被最好的一中拒之门外” there’s a zero whose subject refers to “I”. None of the systems got it. This shows a zero anaphora component would be useful for Chinese-English MT.

  Also, “一中” are two very common characters (meaning “one” and “center”). However, the word that consists of these two characters means “first high school” (lexical issue).

- Sentence 43 (DOC cmn-NG-31-112514-3491242-S1 S1 seg 12)

  Source:

  收到通 知书后父母的心情很差，我一个人出了门，走着走着，天黑了，发现来到了小河边，顺便去宝成伯家看看吧

  Reference:

  After [they//I] received the notice, my parents were very down. I went out by myself and kept walking until dark. I found I had come to the riverbank, and thought I might go and drop a visit to Uncle Baocheng.

  Rosetta segmented source:

  收到 ␣通知书 ␣后 ␣父母 ␣的 ␣心情 ␣很 ␣差 ␣, ␣我 ␣一 ␣个 ␣人出 ␣了 ␣门 ␣, ␣走着 ␣走着 ␣, ␣天黑 ␣了 ␣, ␣发现 ␣来到 ␣了 ␣小 ␣河 ␣边 ␣, ␣顺便 ␣去 ␣宝成 ␣伯家 ␣看 看 ␣吧

  Rosetta:

  After the receipt of of parents feel very bad, I was one of door walked into the night and found that a small river, Chen look.

  Nightingale:

  After the receipt of the notification by parents, I am a person's mood is very poor, walk out the door and walked on, the dark, it was discovered that came to the river, by the way to go to the home to take a look.

  Agile:

  After receiving the parents, I feel very bad out the door, walked on,, found that came to the side, and opportunity to at a look.

  Analysis:

  There are many zeros in the sentence. It might be fixed with good conjunctions (which are not present in the Chinese sentence either).

3.2.1 Summary of Error Analysis

According to our error analysis, we have observed the following:

- Both Agile and Rosetta frequently drop things in translations, but Rosetta does it worse, especially dropping names, numbers, other content words, and whole stretches

towards the end of the sentence.

- Rosetta isn't doing well at reordering NP internal modification syntax. In Chapter 5 we studied DE classification that can help with this issue.
- Rosetta performs a number of inappropriate reorderings of words in translation (rather than good syntactically motivated ones).
- Systems have trouble with topicalized constituents in Chinese.
- There are places where filling in pronoun referents for Chinese zero anaphora would improve systems.
- There are Chinese word segmentation errors which negatively affected translations and might be fixable. In Chapter 2 we discussed what the important characteristics for a good segmentation for MT. Many of the observations were from the analysis in this chapter.
- Agile has a distinctive ability to produce good bits of syntax on the English side (correctly inserting function words, pleonastic subjects, etc.), but in other respects can produce quite bad translations that can mess up whole phrases.
- All systems often had problems figuring out "global structure" or in general how the clauses connect in the long Chinese sentences.
- The quality of the reference sentences is not very good. One obvious example is the wrong translation of the name '戴蒙德' to 'Desmond'. Many of the sentences we presented are from a discussion of the book *Guns, Germs, and Steel: The Fates of Human Societies* by Jared Diamond. Therefore the right translation should be 'Diamond', not 'Desmond'.

  Another example is in the reference translation for the last sentence: "drop a visit to" is non-fluent translation for what should be "drop in on" or just "visit".

In order to provide a quantitative view of our analysis, I also made a table of how often each type of error occurs (Table 3.2). Note the number of "dropping content words"

Error Type	Rosetta	Nightingale	Agile
dropping content words	27	4	11
DE constructions	8	8	6
other Chinese grammatical structures that cause reordering	7	8	7
zero	5	7	7
non-literal expressions or confusing lexical items	7	7	7
unnecessary reordering	2	3	1

Table 3.2: Counts of different error types in the translations of Rosetta, Nightingale, and Agile on the analyzed 24 sentences.

is underestimated for Rosetta. Rosetta tends to drop whole phrases, and when the whole phrase is dropped it only gets counted once. Since we have the segmentation for the Rosetta system, I also checked how many content word drops are due to mis-segmentation. Among the 27 content words dropped, 8 of them are likely due to mis-segmentation. This motivated me to work on improving the segmentation quality.

In Table 3.2, there are two broad categories "other Chinese grammatical structures that cause reordering" and "non-literal expressions or confusing lexical items". Many issues in the first category are addressed in Chapter 4. In my thesis, I did not attempt to address the issues of non-literal expressions and confusing lexical items. Work on using word sense disambiguation techniques to choose better phrases in context (Chan, Ng, and Chiang, Chan et al.; Carpuat and Wu, 2007) can potentially address this category of errors. The category "zero" encompasses when the original Chinese sentences lack the information of a pronoun. This usually occurs in sentences that have several clauses, or when the zero in Chinese refers to an entity in the previous sentence. Therefore, a zero anaphora component that uses sentential and discourse information can help this category of errors. In this thesis I did not address this topic either. The category "unnecessary reordering" shows that sometimes the correct word order to translate is simply the right order, and sometimes the MT systems can perform unnecessary reordering that would mess up the correct translation.

Also note that topicalization is not in the table, because it does not occur frequently in our examples. Even so, whenever topicalization occurs, it is hard for MT systems to get the word order right. There is one case in Sentence 38 where both Rosetta and Agile failed

to translate the topicalization structure correctly.

As a result of this error analysis, I decided to first concentrate on improving the word segmentation quality of the Rosetta system (which is already described in Section 2.3), and then on two of the other problems prominently impacting MT quality: the translation of complex noun phrases involving modifications with DE, and the correct grammatical ordering of phrases when translating from Chinese to English.

Chapter 4

Discriminative Reordering with Chinese Grammatical Relations Features

4.1 Introduction

We can view the machine translation task as consisting of two subtasks: predicting the collection of words in a translation, and deciding the order of the predicted words. These two aspects are usually intertwined during the decoding process. Most systems, phrased-based or syntax-based, score translation hypotheses in their search space with a combination of reordering scores (like distortion penalty or grammar constraints) and lexical scores (like language models). There is also work that focuses on one of the subtasks. For example, Chang and Toutanova (2007) built a discriminative classifier to choose a hypothesis with the best word ordering under an *n*-best reranking framework; Zens and Ney (2006), on the other hand, built a discriminative classifier to classify the orientation of phrases and use it as a component in a phrase-based system.

Based on the analysis in Chapter 3, we know that structural differences between Chinese and English are a major factor in the difficulty of machine translation from Chinese to English. The wide variety of such Chinese-English differences include the ordering of head nouns and relative clauses, and the ordering of prepositional phrases and the heads they modify. Also, in Chinese the character "的" (DE) occurs very often and has ambiguities when mapping into English, which is why we look further into how to classify DE in

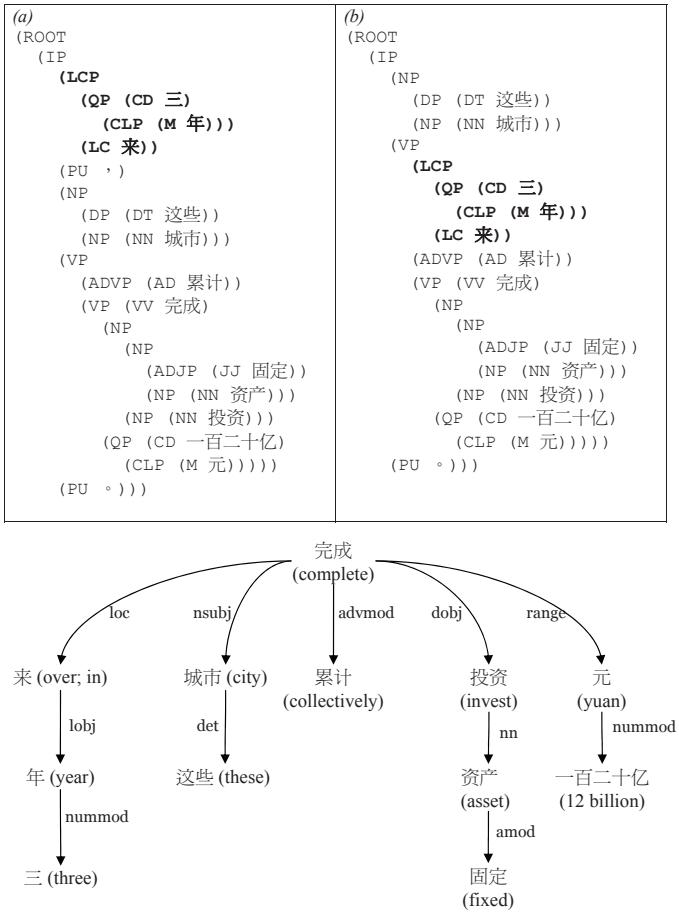


Figure 4.1: Sentences (*a*) and (*b*) have the same meaning, but different phrase structure parses. Both sentences, however, have the same typed dependencies shown at the bottom of the figure.

Chapter 5. The error analysis points in the direction that better understanding of the source language can benefit machine translation.

The machine translation community has spent a considerable amount of effort on using syntax in machine translation. There has been effort on using syntax on the target language side such as Galley et al. (2006); the claim being if the system understands the target language more, it can produce better and more readable output. Previous studies have also shown that using syntactic structures from the source side can help MT performance on these constructions. Most of the previous syntactic MT work has used phrase structure parses in various ways, either by doing syntax directed translation to directly translate parse

trees into strings in the target language (Huang et al., 2006) , or by using source-side CFG or dependency parses to preprocess the source sentences (Wang et al., 2007; Xu et al., 2009).

One intuition for using syntax is to capture different Chinese structures that might have the same meaning and hence the same translation in English. But it turns out that phrase structure (and linear order) are not sufficient to capture this meaning relation. Two sentences with the same meaning can have different phrase structures and linear orders. In the example in Figure 4.1, sentences (*a*) and (*b*) have the same meaning, but different *linear orders* and different *phrase structure parses*. The translation of sentence (*a*) is: "*In the past three years* these municipalities have collectively put together investments in fixed assets in the amount of 12 billion yuan." In sentence (*b*), "in the past three years" has moved its position. The temporal adverbial "三 年 来" (in the past three years) has different linear positions in the sentences. The phrase structures are different too: in (*a*) the `LCP` is immediately under `IP` while in (*b*) it is under `VP`.

We propose to use *typed dependency* parses instead of phrase structure parses. Typed dependency parses give information about grammatical relations between words, instead of constituency information. They capture syntactic relations, such as *nsubj* (nominal subject) and *dobj* (direct object) , but can also encode semantic information such as in the *loc* (localizer) relation. For the example in Figure 4.1, if we look at the sentence structure from the typed dependency parse (bottom of Figure 4.1), "三 年 来" is connected to the main verb 完成 (finish) by a *loc* (localizer) relation, and the structure is the same for sentences (*a*) and (*b*). This suggests that this kind of semantic and syntactic representation could have more benefit than phrase structure parses for MT.

Our Chinese typed dependencies are automatically extracted from phrase structure parses. In English, this kind of typed dependencies has been introduced by de Marneffe et al. (2006) and de Marneffe and Manning (2008). Using typed dependencies, it is easier to read out relations between words, and thus the typed dependencies have been used in meaning extraction tasks.

In this chapter, I use typed dependency parses on the source (Chinese) side to help find better word orders in Chinese-English machine translation. The approach is quite similar to work done at the same time as our work at Google and published as Xu et al. (2009).

Our work differs in using a much richer set of dependencies, such as differentiating different kinds of nominal modification. I hope that this extra detail is helpful in MT, but I have not had a chance to compare the performance of the two dependency representations. This work is also quite similar to the work at Microsoft Research (Quirk et al., 2005) because we are also using dependency parses instead of constituency parses. We are different from (Quirk et al., 2005) because they used unnamed dependencies, but we focus on Chinese typed dependencies (Section 4.3), which are designed to represent the grammatical relations between words in Chinese sentences. Also, our decoding framework is different. Instead of building a different decoding framework like the treelet decoder in Quirk et al. (2005), we design features over the Chinese typed dependencies and use them in a phrase-based MT system when deciding whether one chunk of Chinese words (MT system statistical phrase) should appear before or after another. To achieve this, we train a discriminative phrase orientation classifier following the work by Zens and Ney (2006), and we use the grammatical relations between words as extra features to build the classifier. We then apply the phrase orientation classifier as a feature in a phrase-based MT system to help reordering. We get significant BLEU point gains on three test sets: MT02 (+0.59), MT03 (+1.00) and MT05 (+0.77).[1].

4.2 Discriminative Reordering Model

Basic reordering models in phrase-based systems use linear distance as the cost for phrase movements (Koehn et al., 2003). The disadvantage of these models is their insensitivity to the content and grammatical role of the words or phrases. More recent work (Tillman, 2004; Och et al., 2004; Koehn et al., 2007) has introduced lexicalized reordering models which estimate reordering probabilities conditioned on the actual phrases. Lexicalized reordering models have brought significant gains over the baseline reordering models, but one concern is that data sparseness can make estimation less reliable. Zens and Ney (2006) proposed a discriminatively trained phrase orientation model and evaluated its performance as a classifier and when plugged into a phrase-based MT system. Their framework allows us to easily add in extra features. Therefore, we use it as a testbed to see if we can effectively

[1]This work was first published in (Chang et al., 2009)

use features from Chinese typed dependency structures to help reordering in MT.

4.2.1 Phrase Orientation Classifier

We build up the target language (English) translation from left to right. The phrase orientation classifier predicts the start position of the next phrase in the source sentence. In our work, we use the simplest class definition where we group the start positions into two classes: one class for a position to the left of the previous phrase (*reversed*) and one for a position to the right (*ordered*).

Let $c_{j,j'}$ be the class denoting the movement from source position j to source position j' of the next phrase. The definition is:

$$c_{j,j'} = \begin{cases} \textit{reversed} & \text{if } j' < j \\ \textit{ordered} & \text{if } j' > j \end{cases}$$

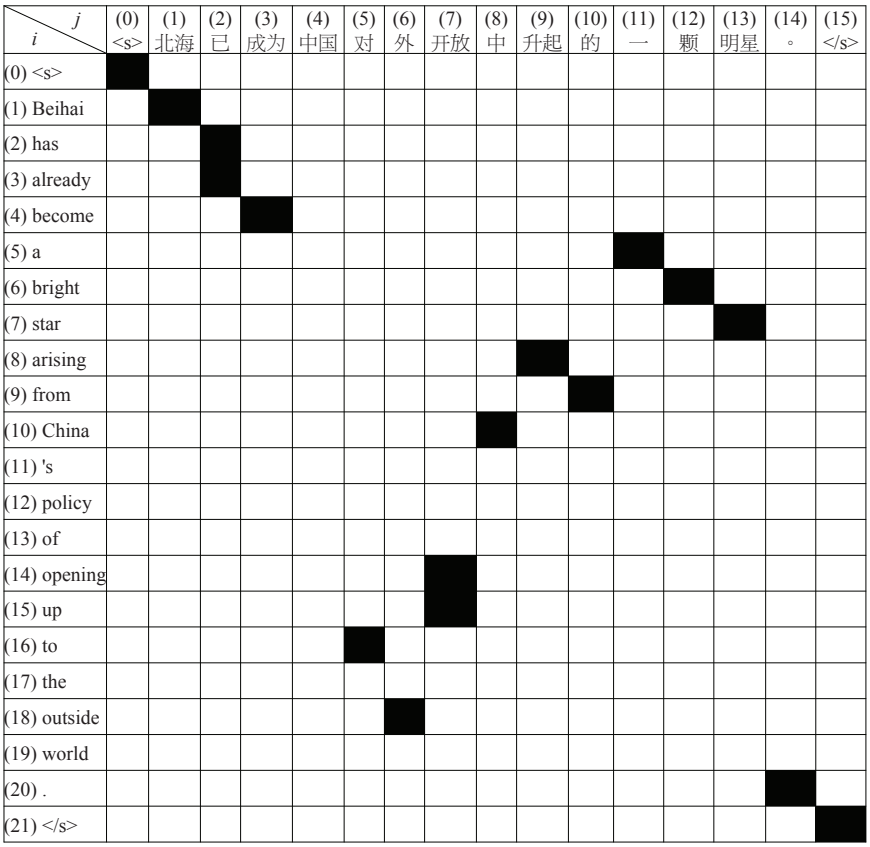
The phrase orientation classifier model is in the log-linear form:

$$p_{\lambda_1^N}(c_{j,j'}|f_1^J, e_1^I, i, j) = \frac{\exp\left(\sum_{n=1}^N \lambda_n h_n(f_1^J, e_1^I, i, j, c_{j,j'})\right)}{\sum_{c'} \exp\left(\sum_{n=1}^N \lambda_n h_n(f_1^J, e_1^I, i, j, c')\right)}$$

i is the target position of the current phrase, and f_1^J and e_1^I denote the source and target sentences respectively. c' represents the two possible categories of $c_{j,j'}$.

We can train this log-linear model on lots of labeled examples extracted from all of the aligned MT training data. Figure 4.2 is an example of an aligned sentence pair and the labeled examples that can be extracted from it. Also, unlike conventional MERT training, we can extract a large number of binary features for the discriminative phrase orientation classifier. The experimental setting will be described in Section 4.4.1.

The basic feature functions we use are similar to what Zens and Ney (2006) used in their MT experiments. The basic binary features are source words within a window of size 3 ($d \in -1, 0, 1$) around the current source position j, and target words within a window of size 3 around the current target position i. In the classifier experiments in Zens and Ney (2006), they also use word classes to introduce generalization capabilities. However, in the MT setting it's harder to incorporate part-of-speech information on the target language.



i	j	j'	class
0	0	1	ordered
1	1	2	ordered
3	2	3	ordered
4	3	11	ordered
5	11	12	ordered
6	12	13	ordered
7	**13**	**9**	**reversed**
8	9	10	ordered
9	**10**	**8**	**reversed**
10	**8**	**7**	**reversed**
15	**7**	**5**	**reversed**
16	5	6	ordered
18	6	14	ordered
20	14	15	ordered

Figure 4.2: An illustration of an alignment grid between a Chinese sentence and its English translation along with the labeled examples for the phrase orientation classifier. Note that the alignment grid in this example is automatically generated.

Zens and Ney (2006) therefore exclude word class information in the MT experiments. In our work we will simply use the word features as basic features for the classification experiments as well. As a concrete example, we look at the labeled example $(i = 4, j = 3, j' = 11)$ in Figure 4.2. We include the word features in a window of size 3 around j and i as in (Zens and Ney, 2006). However, we also include words around j' as features. So we will have nine word features for $(i = 4, j = 3, j' = 11)$:

Src_{-1}:已	Src_0:成为	Src_1:中国
$Src2_{-1}$:的	$Src2_0$:一	$Src2_1$:颗
Tgt_{-1}:already	Tgt_0:become	Tgt_1:a

4.2.2 Path Features Using Typed Dependencies

After parsing a Chinese sentence and extracting its grammatical relations, we design features using the grammatical relations. To predict the ordering of two words, we use the path between the two words annotated by the grammatical relations. Using this feature helps the model learn about what the relation is between the two chunks of Chinese words. The feature is defined as follows: for two words at positions p and q in the Chinese sentence ($p < q$), we find the shortest (undirected) path in the typed dependency parse from p to q, concatenate all the relations on the path and use that as a feature.

A concrete example is the sentence in Figure 4.3, where the alignment grid and labeled examples are shown in Figure 4.2. The glosses of the Chinese words in the sentence are in Figure 4.3, and the English translation is "Beihai has already become a bright star arising from China's policy of opening up to the outside world." which is also listed in Figure 4.2.

For the labeled example $(i = 4, j = 3, j' = 11)$, we look at the typed dependency parse to find the path feature between 成为 and 一. The relevant dependencies are: *dobj*(成为, 明星), *clf*(明星, 颗) and *nummod*(颗, 一). Therefore the path feature is *PATH:dobjR-clfR-nummodR*. We also use the directionality: we add an *R* to the dependency name if it's going against the direction of the arrow. We also found that if we include features of both directions like *dobj* and *dobjR*, these features got incorrectly over-trained, because these features implicitly encode the information of the order of Chinese words. Therefore, we normalized features by only picking one direction. For example, both features *prep-dobjR* and *prepR-dobj* are normalized into the same feature *prep-dobjR*. In other words, if the first relation was reversed, we flip the direction of every relation in the path to normalize the feature. By doing this, the features no longer leak information of the correct class in the training phase, and should be more accurate when used to predict the ordering in the testing phase. So in the case above, the feature will be normalized as *PATH:dobj-clf-nummod*.

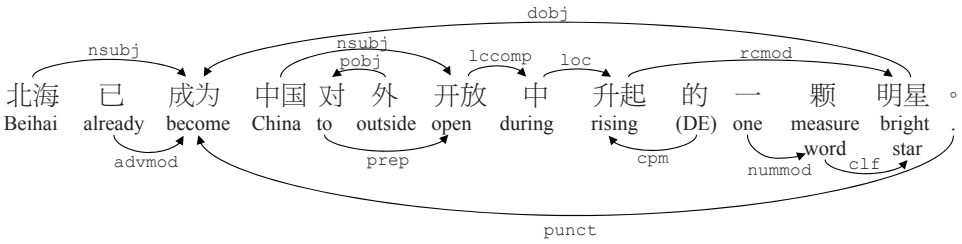


Figure 4.3: A Chinese example sentence labeled with typed dependencies

4.3 Chinese Grammatical Relations

The Chinese typed dependencies are automatically extracted from phrase structure parses. Dependencies and phrase structures (constituency parses) are two different ways of representing sentence structures. A phrase structure is a tree representation of multi-word constituents, where the words are the leaves, and all the other nodes in the tree are either part-of-speech tags or phrasal tags. A dependency parse represents dependency between individual words, and therefore every node in the dependency tree or graph is a word in the sentence. A *typed* dependency parse has additional labels on each dependency between two words that indicate the grammatical relations, such as *subject* or *direct object*. Our Chinese typed grammatical relations closely follow the design principles of the English Stanford typed dependencies (SD) representation (de Marneffe et al., 2006; de Marneffe and Manning, 2008). The goals in designing the Stanford typed dependencies are mostly practical. The hope is to make it easier to apply syntactic structure to all kinds of meaning extraction applications. It is easier to understand because all relationships in a sentence are uniformly described as typed dependencies between pairs of words. We follow the practically oriented design principles of the SD representation to design the Stanford Chinese dependencies. In addition, the Stanford Chinese dependencies try to map the existing English grammatical relations to corresponding Chinese relations as much as possible. The motivation is that multiple languages should be able to convey the same meaning, therefore the meaning representation should be as similar as possible. This also could help cross-lingual applications such as machine translation. There are also some Chinese specific grammatical relations that could not be directly mapped, but could also be useful for applications such as translation. Figure 4.3 is an example Chinese sentence with the typed dependencies between

words. It is straightforward even for a non-linguist to read out relations such as "the nominal subject of *become* is *Beihai*" from the representation.

I will provide descriptions for all 44 Chinese grammatical relations we designed, compare them to the English counterparts, and also give empirical numbers of how often each grammatical relation occurs in Chinese sentences.

4.3.1 Description

There are 44 named grammatical relations, and a default 45th relation *dep* (dependent). If a dependency matches no patterns, it will have the most generic relation *dep*. The dependencies are bi-lexical relations, where a grammatical relation holds between two words: a governor and a dependent. The descriptions of the 44 grammatical relations are listed in alphabetical order. We also get the frequencies of the grammatical relations from files 1–325 in CTB6, and we list the grammatical relations ordered by their frequencies in Table 4.1. The total number of dependencies is 85748, and other than the ones that fall into the 44 grammatical relations, there are also 7470 dependencies (8.71% of all dependencies) that do not match any patterns, and therefore keep the generic name *dep*.

1. ***advmod*: adverbial modifier**

   An adverbial modifier of a word is an ADVP that serves to modify the meaning of the word.

Chinese:	有关	部门	先	送上	这些	法规性	文件
Gloss:	relevant	department	first	send	these	regulatory	document
Translation:	first the appropriate bureau delivers these regulatory documents						
Example dep:	advmod(送上, 先)						

2. ***amod*: adjectival modifier**

   An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of an NP, CLP or QP.

Chinese:	跨世纪	工程
Gloss:	trans-century	engineering
Translation:	a century-spanning undertaking	
Example dep:	amod(工程, 跨世纪)	

3. ***asp*: aspect marker**

   An aspect marker indicates aspect of a verb. The aspect markers have the part-of-speech tag AS, which includes only 了, 着, 过, and 的.

Chinese:	发挥	了	显著	作用
Gloss:	develop	(LE)	significant	role
Translation:	have played a prominent role			
Example dep:	asp(发挥, 了)			

   了 (pronounced: LE) is an aspect marker to indicate past tense.

4. ***assm*: associative marker**

   The part-of-speech tag DEG in Chinese Treebank is a genitive marker or an associative marker. The word with the POS tag DEG (的 or 之) is the associative marker.

Chinese:	浦东	开发	的	有序	进行
Gloss:	Pudong	development	DE	orderly	process
Translation:	the orderly advancement of Pudong's development				
Example dep:	assm(开发, 的)				

5. ***assmod*: associative modifier**

   The part-of-speech tag DEG in Chinese Treebank is a genitive marker or an associative marker. The noun in the phrase with DEG that is used to modify the following phrase, is called the associative modifier.

Chinese:	企业	的	商品
Gloss:	enterprise	DE	commodities
Translation:	the commodities of the enterprise		
Example dep:	assmod(商品, 企业)		

6. ***attr*: attributive**

   The words 是 (be) and 为 (be) are tagged as VC, which means copula verbs. When it's used to link two NPs, the second noun and the VC has an attributive relation.

Chinese:	两	岸	贸易额	为	二百亿	美元
Gloss:	two	shore	volume of trade	be	twenty billion	USD
Translation:	the volume of trade between mainland and Taiwan was 20 billion dollars					
Example dep:	attr(为, 美元)					

7. ***ba*: "ba" construction**

   The *ba*-construction is one of the most widely discussed topics in Chinese linguistics; it does not have a real equivalent in English. In the literature, the word "把/BA" in the *ba*-construction has been argued to be a case marker, a secondary topic marker, a preposition, a verb, and so on (Bender, 2000). In the Penn Chinese Treebank, a unique POS tag *BA* is used.

Chinese:	把	注意力	转向	其他	新兴	市场
Gloss:	BA	attention	shift	other	emerging	market
Translation:	turn their attention to other, newly rising markets					
Example dep:	ba(转向, 把)					

8. ***cc*: coordinating conjunction**

   A coordinating conjunction is the relation between an element and a conjunction.

Chinese:	机械	设备	和	工业	原材料
Gloss:	machine	equipment	and	industry	raw materials
Translation:	machine equipment and industrial raw materials				
Example dep:	cc(原材料, 和)				

9. ***ccomp*: clausal complement**

   In Chinese, clause linkage is not as clear as in English. Therefore the clausal complement is harder to identify. There are more typical cases like:

Chinese:	银行	决定	先	在	日本	取得	信用	评级
Gloss:	bank	decide	first	in	Japan	obtain	credit	rating
Translation:	the bank decided to obtain a credit rating in Japan first							
Example dep:	ccomp(决定, 取得)							

10. ***clf*: classifier modifier**

    A classifier modifier is a relation between the classifier (measure word) and the noun

phrase that it modifies.

Chinese:	七十一	件	法规性	文件
Gloss:	71	pieces (measure word)	regulatory	documents
Translation:	71 regulatory documents			
Example dep:	clf(文件, 件)			

11. ***comod*: coordinated verb compound modifier**

A coordinate verb compound modifier is the relation between verbs under the phrasal tag VCD, which means coordinated verb compound.

Chinese (bracketed):	(VCD	(VV 颁布)	(VV 实行))
Gloss:		promulgate	implement
Example dep:	comod(颁布, 实行)		

12. ***conj*: conjunct (links two conjuncts)**

Conjunct is a relation that links two conjuncts in a coordination structure.

Chinese:	机械	设备	和	工业	原材料
Gloss:	machine	equipment	and	industry	raw materials
Translation:	machine equipment and industrial raw materials				
Example dep:	conj(原材料, 设备)				

13. ***cop*: copular**

A copular (with a POS tag VC) has a *cop* relation with the main verb it's modifying.

Chinese:	原	是	自给自足	的	经济
Gloss:	originally	is	self-sufficient	(DE)	economy
Translation:	the economy originally was self-sufficient				
Example dep:	cop(自给自足, 是)				

14. ***cpm*: complementizer**

Words with the POS tag *DEC*, usually 的 (DE), has a complementizer relation with the head word in the phrase in the CP.

Chinese:	开发	浦东	的	经济	活动
Gloss:	develop	Pudong	(DE)	economy	activity
Translation:	an economic activity in developing Pudong				
Example dep:	cpm(开发, 的)				

15. ***det*: determiner**

A determiner (POS tag *DT*) has a *determiner* relation with the word it modifies.

Chinese:	这些	经济	活动
Gloss:	these	economy	activity
Translation:	these economic activities		
Example dep:	det(活动, 这些)		

16. ***dobj*: direct object**

The direct object of a transitive verb is its accusative object.

Chinese:	浦东	颁布	了	七十一	件	文件
Gloss:	Pudong	promulgate	(LE)	71	piece	documents
Translation:	Pudong has promulgated 71 documents					
Example dep:	dobj(颁布, 文件)					

件 can be translated as "piece", but it is a measure word in Chinese to count documents.

17. ***dvpm*: manner DE(地) modifier**

According to Xia (2000), the part-of-speech tag DEV is a *manner DE*. This only includes 地 when it occurs in "XP 地 VP", where XP modifies the VP. In some old literature, 的 is also used in this pattern so will also be tagged as DEV. The *dvpm* relation is between the 地/DEV or 的/DEV and the word in XP it modifies.

Chinese:	有效	地	防止	了	外汇	流失
Gloss:	effective	(DEV)	prevent	(LE)	foreign exchange	loss
Translation:	effectively preventing the outflow and loss of foreign exchange					
Example dep:	dvpm(有效, 地)					

18. ***dvpmod*: a "XP+DEV(地)" phrase that modifies VP**

The *dvpmod* relation is between a DVP[2] and the phrase it modifies.

In the same example as in *dvpm*, there is a *dvpmod* relation:

Example dep: dvpmod(防止, 有效)

[2] phrase formed by "XP + DEV"

19. ***etc*: etc modifier**

The part-of-speech tag ETC in Chinese is used for the word 等 and 等等.

Chinese:	科技	、	文教	等	领域	
Gloss:	science and technology	,	culture and education	etc	area	
Translation:	areas such as science and technology, culture and education, etc					
Example dep:	etc(文教, 等)					

20. ***lccomp*: clausal complement of a localizer**

In a localizer phrase (LCP), the localizer has a clausal complement. The relation between the localizer and the clausal complement is *lccomp*.

Chinese:	中国	对	外	开放	中	升起	的	明星
Gloss:	China	towards	outside	open	amidst	rising	DE	bright star
Translation:	a bright star arising from China's policy of opening up to the outside world							
Example dep:	lccomp(中, 开放)							

21. ***loc*: localizer**

The localizer relation is between the localizer and the VP or IP it modifies.

```
(VP (VV 占)
    (LCP
      (QP (CD 九成))
      (LC 以上)))
```

Example dep: loc(占, 以上)

Here is another example where the LCP comes before the VP:

```
(VP
  (LCP
    (NP (NN 经济) (NN 合作区))
    (LC 内))
  (ADVP (AD 已))
  (VP
    (VP (VV 开发)
      (QP (CD 二十二点六)
        (CLP (M 平方公里))))))
```

Example dep: loc(开发, 内)

22. ***lobj*: localizer object**

    Localizer object of a localizer is the main noun in the LCP before LC.

```
(LCP
  (NP (NT 近年))
    (LC 来))
```

Example dep: lobj(来, 近年)

23. ***mmod*: modal verb modifier**

    A modal verb modifier is when a VP contains a VV followed by a VP. We call the first VV a modal verb modifier of the second VP.

```
(IP
  (NP (NN 利益))
  (VP (VV 能)
    (VP (VV 得到)
      (NP (NN 保障)))))
```

Example dep: mmod(得到, 能)

24. ***neg*: negative modifier**

    When a verb has a negative modifier, its meaning is negated. The negative modifier is the character 不/AD in Chinese, which means “no” in English.

Chinese:	以前	不	曾	遇到	过
Gloss:	past	no	already	encounter	(aspect marker for the past)
Translation:	have not been encountered before				
Example dep:	neg(遇到, 不)				

25. ***nn*: noun compound modifier**

In a NP with multiple nouns, the head is the last noun and every previous noun is a noun compound modifier of it.

Chinese:	药品	采购	服务	中心
Gloss:	drug	procurement	service	center
Translation:	drug purchase service center			
Example dep:	nn(中心, 药品), nn(中心, 采购), nn(中心, 服务)			

26. ***nsubjpass*: nominal passive subject**

The nominal passive subject is a subject of a passive clause. The passive marker in Chinese is 被/SB.

Chinese:	镍	被	称作	现代	工业	的	维生素
Gloss:	Nickel	(SB)	called	modern	industry	(DE)	vitamin
Translation:	Nickel is called the vitamin of modern industry						
Example dep:	nsubjpass(称作, 镍)						

27. ***nsubj*: nominal subject**

A nominal subject is a noun phrase which is the syntactic subject of a clause.

Chinese:	梅花	盛开
Gloss:	plum flowers	bloom
Translation:	The plum flowers bloom.	
Example dep:	nsubj(盛开, 梅花)	

28. ***nummod*: number modifier**

A number modifier is a relation between a number and the noun it modifies.

Chinese:	浦东	颁布	了	七十一	件	文件
Gloss:	Pudong	promulgate	(LE)	71	piece	documents
Translation:	Pudong has promulgated 71 documents					
Example dep:	nummod(件, 七十一)					

29. ***ordmod*: ordinal number modifier**

An ordinal number modifier is a relation between an ordinal number and the noun it modifies.

Chinese:	第七	个	海关	机构
Gloss:	7th	(measure word)	customs	organization
Translation:	the seventh customs organization			
Example dep:	ordmod(个, 第七)			

30. ***pass*: passive marker**

When the verb has a passive marker (被, part-of-speech tags SB or LB) modifying it, there is a passive marker relation between them.

Chinese:	被	认定	为	高	技术	产业
Gloss:	(SB)	consider	as	high	technology	industry
Translation:	have been identified as high level technology enterprises					
Example dep:	pass(认定, 被)					

31. ***pccomp*: clausal complement of a preposition**

A clausal complement of a preposition is a relation between a clausal complement and the preposition that introduces it. For example,

```
(PP (P 因为)
      (IP
        (VP
          (VP
            (ADVP (AD 一))
            (VP (VV 开始)))
          (VP
            (ADVP (AD 就))
            (ADVP (AD 比较))
            (VP (VA 规范)))))))
```

Example dep: pccomp(因为, 开始)

32. ***plmod*: localizer modifier of a preposition**

When a preposition comes in front of an LCP, there exists a relation of the localizer

modifying the preposition.

```
(PP (P 在)
    (LCP
      (NP
        (DP (DT 这)
            (CLP (M 片)))
        (NP (NN 热土)))
      (LC 上)))
```

Example dep: plmod(在, 上)

33. ***pobj*: prepositional object**

The prepositional object of a preposition is the noun that is its object.

```
(PP (P 根据)
      (NP
        (DNP
          (NP
            (NP (NN 国家))
            (CC 和)
            (NP (NR 上海市)))
          (DEG 的))
        (ADJP (JJ 有关))
        (NP (NN 规定))))
```

Example dep: pobj(根据, 规定)

34. ***prep*: prepositional modifier**

A prepositional modifier of an NP or a VP is any prepositional phrase that modifies the meaning of it.

```
(IP
  (VP
    (PP (P 在)
      (LCP
        (NP (NN 实践))
        (LC 中)))
    (ADVP (AD 逐步))
    (VP (VV 完善))))
```

Translation: awaiting step-by-step completion as they are put into practice
Example dep: prep(完善, 在)

35. ***prnmod*: parenthetical modifier**

When parentheses appear, the phrase inside the parentheses is a parenthetical modifier of the noun phrase it modifies.

Chinese:	八五	期间	（	一九九 0 年	–	一九九五年	）
Gloss:	eighty five-year plan	period	(	1990	–	1995	)

Translation: "eighty five-year plan" period (1990 – 1995))
Example dep: prnmod(期间, 一九九五年)

36. ***prtmod*: particles such as 所, 以, 来, 而**

The particle verb relation is between a particle (part-of-speech tag MSP) and the verb it modifies.

Chinese:	在	产业化	所	取得	的	成就
Gloss:	at	industrialization	(MSP)	obtain	(DE)	achievement

Translation: the achievements we have made in industrialization
Example dep: prtmod(取得, 所)

37. ***punct*: punctuation**

This is used for any piece of punctuation in a clause, if punctuation is being retained in the typed dependencies.

Chinese:	统计	表明	，	进出口	呈	上升	之	势
Gloss:	statistics	show	,	import and export	show	rising		trend

Translation: The statistics showed that the import and export is on the rise.

Example dep: punct(表明, ，)

38. ***range*: dative object that is a quantifier phrase**

Range is the indirect object of a VP that is the quantifier phrase which is the (dative) object of the verb.

Chinese:	成交	药品	一亿多	元
Gloss:	conclude a transaction	drugs	more than 100 million	yuan

Translation: concluded translations for drugs of over 100 million yuan

Example dep: range(成交, 元)

39. ***rcmod*: relative clause modifier**

The relative modifier is the CP that modifies an NP.

```
(NP
  (CP
    (IP
      (VP
        (ADVP (AD 不))
        (ADVP (AD 曾))
        (VP (VV 遇到) (AS 过))))
    (DEC 的))
  (NP
    (NP (NN 情况))))
```

Translation: situations that have not been encountered

Example dep: rcmod(情况, 遇到)

40. ***rcomp*: resultative complement**

In the Penn Chinese Treebank, the phrasal category VRD is the verb-resultative and verb-directional compounds, where there are two distinctive constituents with the

second constituent indicating the direction result of the first constituent. The second verb is a resultative complement of the first one.

```
(VP
  (VRD (VV 研究) (VA 成功)))
```

Example dep: rcomp(研究, 成功)

41. ***tmod*: temporal modifier**

    Temporal modifier is a temporal noun and the VP it modifies.

```
(IP
  (VP
    (NP (NT 以前))
    (ADVP (AD 不))
    (ADVP (AD 曾))
    (VP (VV 遇到) (AS 过))))
```

Example dep: tmod(遇到, 以前)

42. ***top*: topic**

    When the verb is a VC (copula) or a VE (有, similar to "there is" in English) the subject is the "topic" of the verb.

```
(IP
  (NP (NN 建筑))
  (VP (VC 是)
    (NP
      (CP
        (IP
          (VP (VV 开发)
            (NP (NR 浦东))))
        (DEC 的))
      (QP (CD 一)
        (CLP (M 项)))
      (ADJP (JJ 主要))
      (NP (NN 经济) (NN 活动)))))
```

Example dep: top(是, 建筑)

43. ***vmod*: verb modifier**

When a NP has an embedded IP modifier, the main verb of the IP is a verb modifier of the NP.

```
(NP
  (NP (PN 其))
  (DNP
    (PP (P 在)
      (NP
        (IP
          (VP (VV 支持)
            (NP (NN 外商) (NN 投资) (NN 企业))))
        (NP (NN 方面))))
    (DEG 的))
  (NP (NN 主渠道) (NN 作用)))
```

Example dep: vmod(方面, 支持)

44. ***xsubj*: controlling subject**

The controlling subject of a verb is the external subject.

Chinese:	银行	决定	先	在	日本	取得	信用	评级
Gloss:	bank	decide	first	in	Japan	obtain	credit	rating
Translation:	the bank decided to obtain a credit rating in Japan first							
Example dep:	xsubj(取得, 银行)							

4.3.2 Chinese Specific Structures

Although we designed the typed dependencies to show structures that exist both in Chinese and English, there are many other syntactic structures that only exist in Chinese. The typed dependencies we designed also cover those Chinese specific structures. For example, the usage of "的" (DE) is one thing that could lead to different English translations. In the Chinese typed dependencies, there are relations such as *cpm* (DE as complementizer) or *assm* (DE as associative marker) that are used to mark these different structures. The Chinese-specific "把" (BA) construction also has a relation *ba* dedicated to it.

The typed dependencies annotate these Chinese specific relations, but do not directly provide a mapping onto how they are translated into English. Applying the typed dependencies as features in the phrase orientation classifier makes the effect of the reordering from Chinese to English more obvious. This will be further discussed in Section 4.4.4.

4.3.3 Comparison with English Typed Dependencies

To compare the distribution of Chinese typed dependencies with English, we extracted the English typed dependencies from the translation of files 1–325 in the English Chinese Translation Treebank 1.0 (LDC2007T02), which correspond to files 1–325 in CTB6. The English typed dependencies are extracted using the Stanford Parser.

There are 116,799 total English dependencies, and 85,748 Chinese ones. On the corpus we use, there are 44 distinct dependency types (not including *dep*) in Chinese, and 50 in English. The coverage of named relations is 91.29% in Chinese and 90.48% in English; the remainder are the unnamed relation *dep*. We looked at the 18 shared relations between Chinese and English in Table 4.2. Chinese has more *nn*, *punct*, *nsubj*, *rcmod*, *dobj*, *advmod*, *conj*, *nummod*, *attr*, *tmod*, and *ccomp* while English uses more *pobj*, *det*, *prep*, *amod*, *cc*,

cop, and *xsubj*, due mainly to grammatical differences between Chinese and English. For example, some determiners in English (e.g., "the" in (1b)) are not mandatory in Chinese:

(1a) 进出口/import and export 总额/total value

(1b) The total value of imports and exports

In another difference, English uses adjectives (*amod*) to modify a noun ("financial" in (2b)) where Chinese can use noun compounds ("金融/finance" in (2a)).

(2a) 西藏/Tibet 金融/finance 体制/system 改革/reform

(2b) the reform in Tibet 's financial system

We also noticed some larger differences between the English and Chinese typed dependency distributions. We looked at specific examples and provide the following explanations.

prep* and *pobj English has many more uses of *prep* and *pobj*. We examined the data and found three major reasons:

1. Chinese uses both prepositions and postpositions while English only has prepositions. "After" is used as a postposition in Chinese example (3a), but a preposition in English (3b):

   (3a) 九七/1997 之後/after

   (3b) after 1997

2. Chinese uses noun phrases in some cases where English uses prepositions. For example, "之间" (period, or during) is used as a noun phrase in (4a), but it's a preposition in English.

   (4a) 九七/1997 到/to 九八/1998 之间 /period

   (4b) during 1997-1998

3. Chinese can use noun phrase modification in situations where English uses prepositions. In example (5a), Chinese does not use any prepositions between "apple company" and "new product", but English requires use of either "of" or "from".

   (5a) 苹果公司/apple company 新产品/new product

   (5b) the new product of (or from) Apple

   The Chinese DE constructions are also often translated into prepositions in English.

cc* and *punct The Chinese sentences contain more punctuation (*punct*) while the English translation has more conjunctions (*cc*), because English uses conjunctions to link clauses ("and" in (6b)) while Chinese tends to use only punctuation ("," in (6a)).

(6a) 这些/these 城市/city 社会/social 经济/economic 发展/development 迅速/rapid ，
地方/local 经济/economic 实力/strength 明显/clearly 增强/enhance

(6b) In these municipalities the social and economic development has been rapid, and the local economic strength has clearly been enhanced

rcmod* and *ccomp There are more *rcmod* and *ccomp* in the Chinese sentences and fewer in the English translation, because of the following reasons:

1. Some English adjectives act as verbs in Chinese. For example, 新 (new) is an adjectival predicate in Chinese and the relation between 新 (new) and 制度 (system) is *rcmod*. But "new" is an adjective in English and the English relation between "new" and "system" is *amod*. This difference contributes to more *rcmod* in Chinese.

   (7a) 新/new 的/(DE) 核销/verify and write off
   (7b) a new sales verification system

2. Chinese has two special verbs (VC): 是 (SHI) and 为 (WEI) which English doesn't use. For example, there is an additional relation, *ccomp*, between the verb 是/(SHI) and 降低/reduce in (8a). The relation is not necessary in English, since 是/SHI is not translated.

   (8a) 二/second 是/(SHI) 一九九六年/1996
   中国/China 大幅度/substantially
   降低/reduce 关税/tariff

   (8b) Second, China reduced tax substantially in 1996.

conj There are more *conj* in Chinese than in English for three major reasons. First, sometimes one complete Chinese sentence is translated into several English sentences. Our *conj* is defined for two grammatical roles occurring in the same sentence, and therefore, when a sentence breaks into multiple ones, the original relation does not apply. Second, we define

the two grammatical roles linked by the *conj* relation to be in the same word class. However, words which are in the same word class in Chinese may not be in the same word class in English. For example, adjective predicates act as verbs in Chinese, but as adjectives in English. Third, certain constructions with two verbs are described differently between the two languages: verb pairs are described as coordinations in a serial verb construction in Chinese, but as the second verb being the complement of the first verb in English.

4.4 Experimental Results

4.4.1 Experimental Setting

We use various Chinese-English parallel corpora, including LDC2002E18, LDC2003E07, LDC2003E14, LDC2005E83, LDC2005T06, LDC2006E26, LDC2006E85, LDC2002L27 and LDC2005T34, for both training the phrase orientation classifier and for extracting statistical phrases for the phrase-based MT system. The parallel data contain $1,560,071$ sentence pairs from various parallel corpora. There are $12,259,997$ words on the English side. Chinese word segmentation is done by the Stanford Chinese segmenter (Chang et al., 2008). After segmentation, there are 11,061,792 words on the Chinese side. The word alignment is done by the Berkeley word aligner (Liang et al., 2006) and then symmetrized using the grow-diag heuristic.

For the phrase orientation classifier experiments, we extracted labeled examples using the parallel data and the alignment as in Figure 4.2. We extracted $9,194,193$ total valid examples: 86.09% of them are *ordered* and the other 13.91% are *reversed*. To evaluate the classifier performance, we split these examples into training, dev and test set $(8:1:1)$. The phrase orientation classifier used in MT experiments is trained with all of the available labeled examples. In order to train dependency features, we parsed all Chinese side of the parallel data with the Chinese version of the Stanford parser (Levy and Manning, 2003) and automatically converted the result to dependencies.

Our MT experiments use a re-implementation of Moses (Koehn et al., 2003) called *Phrasal*, which provides an easier API for adding features. We use a 5-gram language model trained on the Xinhua and AFP sections of the Gigaword corpus (LDC2007T40)

and also the English side of all the LDC parallel data permissible under the NIST08 rules. Documents of Gigaword released during the epochs of MT02, MT03, MT05, and MT06 were removed. For features in MT experiments, we incorporate Moses's standard eight features as well as the lexicalized reordering features. To have a more comparable setting with (Zens and Ney, 2006), we also have a baseline experiment with only the standard eight features. Parameter tuning is done with Minimum Error Rate Training (MERT) (Och, 2003). The tuning set for MERT is the NIST MT06 data set, which includes 1664 sentences. We evaluate the result with MT02 (878 sentences), MT03 (919 sentences), and MT05 (1082 sentences).

4.4.2 Phrase Orientation Classification Experiments

The basic source word features described in Section 4.2 are referred to as Src and the target word features as Tgt. The feature set that Zens and Ney (2006) used in their MT experiments is Src+Tgt. In addition to that, we also experimented with source word features Src2 which are similar to Src, but take a window of 3 around j' instead of j. In Table 4.3 we can see that adding the Src2 features increased the total number of features by almost 50%, but also improved the performance. The PATH features add fewer total number of features than the lexical features, but still provide a 10% error reduction and 1.63 on the macro-F1 on the dev set. We use the best feature set from the feature engineering in Table 4.3 and test it on the test set. We get 96.38% accuracy and 92.10 macro-F1. The overall improvement of accuracy over the baseline is 10.09 absolute points.

4.4.3 MT Experiments

In the MT setting, we use the log probability from the phrase orientation classifier as an extra feature. For phrases that are in "ordered" orientation, we used the score from $\log P(\text{ordered}|\text{phrases})$, and for those in "reversed" orientation, we used the score from $\log P(\text{reversed}|\text{phrases})$. The weight of this discriminative reordering feature is also tuned by MERT, along with other Moses features. In order to understand how much the PATH features add value to the MT experiments, we trained two phrase orientation classifiers with different features: one with the Src+Src2+Tgt feature set, and the other one with

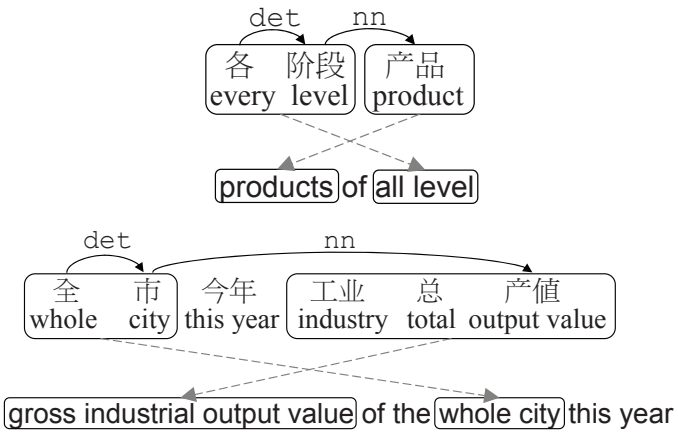


Figure 4.4: Two examples for the feature *PATH:det-nn* and how the reordering occurs.

Src+Src2+Tgt+PATH. The results are listed in Table 4.4. We compared to two different baselines: one is Moses9Feats which has a distance-based reordering model, the other is Baseline, which also includes lexicalized reordering features. From the table we can see that using the discriminative reordering model with PATH features gives significant improvement over both baselines. If we use the discriminative reordering model without PATH features and only with word features, we still get improvement over the Moses9Feats baseline, but the MT performance is not significantly different from Baseline, which uses lexicalized reordering features. We think the reason is that the discriminative reordering model with only the lexical features is not getting much more information than the lexicalized models in Baseline, since they are both only using lexical information. Once that PATH features were added in, the discriminative model became more powerful and was able to outperform Baseline. From Table 4.4 we see that using the Src+Src2+Tgt+PATH features significantly outperforms both baselines. Also, if we compare between Src+Src2+Tgt and Src+Src2+Tgt+PATH, the differences are also statistically significant, which shows the effectiveness of the path features.

4.4.4 Analysis: Highly-weighted Features in the Phrase Orientation Model

There are many features in the log-linear phrase orientation model. We looked at some highly-weighted PATH features to understand what kinds of grammatical constructions were informative for phrase orientation. We found that many path features corresponded to our intuitions. For example, the feature *PATH:prep-dobjR* has a high weight for being *reversed*. This feature informs the model that in Chinese a PP usually appears before VP, but in English they should be reversed. Other features with high weights include features related to the DE construction that is more likely to translate to a relative clause, such as *PATH:advmod-rcmod* and *PATH:rcmod*. They also indicate that the phrases are more likely to be chosen in reversed order. Another frequent pattern that has not been emphasized in the previous literature is *PATH:det-nn*, meaning that a [DT $NP_1 NP_2$] in Chinese is translated into English as [NP_2 DT NP_1]. Examples with this feature are in Figure 4.4. We can see that the important features decided by the phrase orientation model are also important from a linguistic perspective.

4.4.5 Analysis: MT Output Sentences

In addition to the examples in Section 4.4.4 that showed how specific PATH features helped the MT quality, we also took some examples from one of our test sets (MT02) to show examples of full sentences that were improved or worsened by the discriminative reordering models with PATH features. In the following examples, "Baseline system output" refers to the "Baseline (Moses with lexicalized reordering)" column in Table 4.4, and "Improved system output" refers to the "Baseline+DiscrimRereorderWithPATH" column. The reference sentence is one of the four references provided by NIST evaluation. All the MT system outputs and the reference here are uncased.

The first example is the following sentence:

Source:

乌共 领导人 西蒙 年科 同 一 天 说 , 美国 的 行为 是 对 乌 内政 的 粗暴 干涉 , 是 对 乌 国 家 独立 的 威胁 。

Reference:

leader of the ukrainian communist party symonenko said on the same day that the us act was brazen interference in ukraine 's internal affairs and a threat to ukraine 's independence .

Baseline system output:

the ukrainian leader petr simonenko said on the same day , ukraine , the united states is a brutal interference in the internal affairs of an independent state is a threat .

Improved system output:

the ukrainian leader petr simonenko said on the same day , the united states is a brutal interference in the internal affairs of ukraine is a threat to the independence of ukraine .

In this example, the two DE constructions "对 乌 内政 的 粗暴 干涉" (a brutal interference in the internal affairs of ukraine) and "对 乌 国 家 独立 的 威胁" (a threat to the independence of ukraine) were both is captured in the improved system, but not in the baseline system. However, none of the systems captured how the last clause linked to the other part of the sentence. In this example, the last clause linkage should be a coordination.

The second example is as follows:

Source:

他们 来自 六 个 家庭 , 其中 两 个 女孩 没有 父母 。

Reference:

they come from 6 families and 2 are orphaned girls .

Baseline system output:

they were from a family of six , including two girls without parents .

Improved system output:

they were from a family of six , including the parents of the two girls who did not .

This examples shows the improved system decided to reorder the last noun "父母" (parents) to before "两 个 女孩" (two girls), which resulted in a worse translation than the original baseline output.

4.5 Conclusion

We introduced a set of Chinese typed dependencies that gives information about grammatical relations between words, and which may be useful in other NLP applications as well as MT. We used the typed dependencies to build path features and used them to improve a phrase orientation classifier. The path features gave a 10% error reduction on the accuracy of the classifier and 1.63 points on the macro-F1 score. We applied the log probability as an additional feature in a phrase-based MT system, which improved the BLEU score of the three test sets significantly (0.59 on MT02, 1.00 on MT03 and 0.77 on MT05). This shows that typed dependencies on the source side are informative for the reordering component in a phrase-based system. Whether typed dependencies can lead to improvements in other syntax-based MT systems remains a question for future research. The Chinese typed dependencies might be appropriate to integrate in systems that already make use of source side dependencies. The formulation would have to change to take into account the type information.

abbreviation	short description	counts	percentage
nn	noun compound modifier	13278	15.48%
punct	punctuation	10896	12.71%
nsubj	nominal subject	5893	6.87%
conj	conjunct (links two conjuncts)	5438	6.34%
dobj	direct object	5221	6.09%
advmod	adverbial modifier	4231	4.93%
prep	prepositional modifier	3138	3.66%
nummod	number modifier	2885	3.36%
amod	adjectival modifier	2691	3.14%
pobj	prepositional object	2417	2.82%
rcmod	relative clause modifier	2348	2.74%
cpm	complementizer	2013	2.35%
assm	associative marker	1969	2.30%
assmod	associative modifier	1941	2.26%
cc	coordinating conjunction	1763	2.06%
clf	classifier modifier	1558	1.82%
ccomp	clausal complement	1113	1.30%
det	determiner	1113	1.30%
lobj	localizer object	1010	1.18%
range	dative object that is a quantifier phrase	891	1.04%
asp	aspect marker	857	1.00%
tmod	temporal modifier	679	0.79%
plmod	localizer modifier of a preposition	630	0.73%
attr	attributive	534	0.62%
mmod	modal verb modifier	497	0.58%
loc	localizer	428	0.50%
top	topic	380	0.44%
pccomp	clausal complement of a preposition	374	0.44%
etc	etc modifier	295	0.34%
lccomp	clausal complement of a localizer	207	0.24%
ordmod	ordinal number modifier	199	0.23%
xsubj	controlling subject	192	0.22%
neg	negative modifier	186	0.22%
rcomp	resultative complement	176	0.21%
comod	coordinated verb compound modifier	150	0.17%
vmod	verb modifier	133	0.16%
prtmod	particles such as 所, 以, 来, 而	124	0.14%
ba	"ba" construction	95	0.11%
dvpm	manner DE(地) modifier	73	0.09%
dvpmod	a "XP+DEV(地)" phrase that modifies VP	69	0.08%
prnmod	parenthetical modifier	67	0.08%
cop	copular	59	0.07%
pass	passive marker	53	0.06%
nsubjpass	nominal passive subject	14	0.02%

Table 4.1: Chinese grammatical relations and distributions. The counts are from files 1–325 in CTB6.

Shared relations	Chinese	English
nn	15.48%	6.81%
punct	12.71%	9.64%
nsubj	6.87%	4.46%
rcmod	2.74%	0.44%
dobj	6.09%	3.89%
advmod	4.93%	2.73%
conj	6.34%	4.50%
num/nummod	3.36%	1.65%
attr	0.62%	0.01%
tmod	0.79%	0.25%
ccomp	1.30%	0.84%
xsubj	0.22%	0.34%
cop	0.07%	0.85%
cc	2.06%	3.73%
amod	3.14%	7.83%
prep	3.66%	10.73%
det	1.30%	8.57%
pobj	2.82%	10.49%

Table 4.2: The percentage of typed dependencies in files 1–325 in Chinese (CTB6) and English (English-Chinese Translation Treebank)

Feature Sets	#features	Train. Acc. Acc. (%)	Train. Macro-F	Dev Acc. (%)	Dev Macro-F	Test Acc. (%)	Test Macro-F
Majority class	-	86.09	-	86.09	-	-	-
Src	1483696	89.02	71.33	88.14	69.03	-	-
Src+Tgt	2976108	92.47	82.52	91.29	79.80	-	-
Src+Src2+Tgt	4440492	95.03	88.76	93.64	85.58	-	-
Src+Src2+Tgt+PATH	4674291	97.74	95.13	96.35	92.09	96.38	92.10

Table 4.3: Feature engineering of the phrase orientation classifier. Accuracy is defined as (#correctly labeled examples) divided by (#all examples). The macro-F is an average of the accuracies of the two classes. We only used the best set of features on the test set. The overall improvement of accuracy over the baseline is 10.09 absolute points.

Setting	#MERT features	MT06(tune)	MT02	MT03	MT05
Moses9Feats	9	31.49	31.63	31.26	30.26
Moses9Feats+DiscrimRereorderNoPATH	10	31.76(+0.27)	31.86(+0.23)	**32.09**(+**0.83**)	**31.14**(+**0.88**)
Moses9Feats+DiscrimRereorderWithPATH	10	**32.34**(+**0.85**)	**32.59**(+**0.96**)	**32.70**(+**1.44**)	**31.84**(+**1.58**)
Baseline (Moses with lexicalized reordering)	16	32.55	32.56	32.65	31.89
Baseline+DiscrimRereorderNoPATH	17	32.73(+0.18)	32.58(+0.02)	32.99(+0.34)	31.80(−0.09)
Baseline+DiscrimRereorderWithPATH	17	**32.97**(+**0.42**)	**33.15**(+**0.59**)	**33.65**(+**1.00**)	**32.66**(+**0.77**)

Table 4.4: MT experiments of different settings on various NIST MT evaluation datasets. All differences marked in bold are significant at the level of 0.05 with the approximate randomization test in Riezler and Maxwell (2005).

Chapter 5

Disambiguating "DE"s in Chinese

5.1 Introduction

Structural differences between Chinese and English, such as the different orderings of head nouns and relative clauses, cause a great difficulty in Chinese-English MT reflected in the consistently lower BLEU scores than those seen in other difficult language pairs like Arabic-English. Many of these structural differences are related to the ubiquitous Chinese 的 (DE) construction, used for a wide range of noun modification constructions (both single word and clausal) and other uses. Part of the solution to dealing with these ordering issues is hierarchical decoding, such as the Hiero system (Chiang, 2005), a method motivated by 的 (DE) examples like the one in Figure 5.1. In this case, the translation goal is to rotate

澳洲	是	与	北韩	有	邦交	的	少数	国家	之一	。
Aozhou	shi	yu	Beihan	you	bangjiao	DE	shaoshu	guojia	zhiyi	.
Australia	is	with	North Korea	have	diplomatic relations	that	few	countries	one of	.

'Australia is one of the few countries that have diplomatic relations with North Korea.'

Figure 5.1: An example of the DE construction from (Chiang, 2005)

the noun head and the preceding relative clause around 的 (DE), so that we can translate to "[one of few countries] 的 [have diplomatic relations with North Korea]". Hiero can learn this kind of lexicalized synchronous grammar rule.

However, use of hierarchical decoders has not solved the DE construction translation problem. In Chapter 3, we analyzed the errors of three state-of-the-art systems (the 3

DARPA GALE phase 2 teams' systems), and even though all three use some kind of hierarchical system, we found many remaining errors related to reordering. One is shown again here:

当地 一所 名声不佳 的 中学
local a bad reputation DE middle school
Reference: 'a local middle school with a bad reputation'
Team 1: 'a bad reputation of the local secondary school'
Team 2: 'the local a bad reputation secondary school'
Team 3: 'a local stigma secondary schools'

None of the teams reordered "bad reputation" and "middle school" around the 的 (DE). We argue that this is because it is not sufficient to have a formalism which *supports* phrasal reordering, it is also necessary to have sufficient linguistic modeling that the system *knows when and how much to rearrange*.

An alternative way of dealing with structural differences is to reorder source language sentences to minimize structural divergence with the target language (Xia and McCord, 2004; Collins et al., 2005; Wang et al., 2007). For example Wang et al. (2007) introduced a set of rules to decide if a 的 (DE) construction should be reordered or not before translating to English:

- For DNPs (consisting of"XP+DEG"):
  - Reorder if XP is PP or LCP;
  - Reorder if XP is a non-pronominal NP
- For CPs (typically formed by "IP+DEC"):
  - Reorder to align with the "that+clause" structure of English.

Although this and previous reordering work has led to significant improvements, errors still remain. Indeed, Wang et al. (2007) found that the precision of their NP rules is only about 54.6% on a small human-judged set.

One possible reason the 的 (DE) construction remains unsolved is that previous work has paid insufficient attention to the many ways the 的 (DE) construction can be translated,

and the rich structural cues to the translation. Wang et al. (2007), for example, characterized 的 (DE) into only two classes. But our investigation shows that there are many strategies for translating Chinese [A 的 B] phrases into English, including the patterns in Table 5.1, only some involving reversal.

Notice that the presence of reordering is only one part of the rich structure of these examples. Some reorderings are relative clauses, while others involve prepositional phrases, but not all prepositional phrase uses involve reorderings. These examples suggest that capturing finer-grained translation patterns could help achieve higher accuracy both in reordering and in lexical choice.

In this chapter, we propose to use a statistical classifier trained on various features to predict for a given Chinese 的 (DE) construction both whether it will reorder in English and which construction it will translate to in English. We suggest that the necessary classificatory features can be extracted from Chinese, rather than English. The 的 (DE) in Chinese has a unified meaning of 'noun modification', and the choice of reordering and construction realization are mainly a consequence of facts in English noun modification. Nevertheless, most of the features that determine the choice of a felicitous translation are available in the Chinese source. Noun modification realization has been widely studied in English (e.g., (Rosenbach, 2003)), and many of the important determinative properties (e.g., topicality, animacy, prototypicality) can be detected working in the source language.

We first present some corpus analysis characterizing different DE constructions based on how they get translated into English (Section 5.2). We then train a classifier to label DEs into the 5 different categories that we define (Section 5.3). The fine-grained DEs, together with reordering, are then used as input to a statistical MT system (Section 5.5). We find that classifying DEs into finer-grained tokens helps MT performance, usually at least twice as much as just doing phrasal reordering[1]

5.2 DE classification

The Chinese character DE serves many different purposes. According to the Chinese Treebank tagging guidelines (Xia, 2000), the character can be tagged as DEC, DEG, DEV, SP,

[1]This work was first published in Chang et al. (2009).

DER, or AS. Similarly to (Wang et al., 2007), we only consider the majority case when the phrase with 的 (DE) is a noun phrase modifier. The DEs in NPs have a part-of-speech tag of DEC (a complementizer or a nominalizer) or DEG (a genitive marker or an associative marker).

5.2.1 Class Definition

The way we categorize the DEs is based on their behavior when translated into English. This is implicitly done in the work of Wang et al. (2007) where they use rules to decide if a certain DE and the words next to it will need to be reordered. In this work, we categorize DEs into finer-grained categories. For a Chinese noun phrase `[A 的 B]`, we categorize it into one of these five classes:

1. `A B`

   In this category, `A` in the Chinese side is translated as a pre-modifier of B. In most of the cases, `A` is an adjective form, like Example 1.1 in Table 5.1 or the possessive adjective example in Example 1.2. Compound nouns where `A` becomes a pre-modifier of `B` also fit in this category (Example 1.3).

2. `B` *preposition* `A`

   There are several cases that get translated into the form `B` *preposition* `A`. For example, the *of*-genitive in Example 2.1 in Table 5.1.

   Example 2.2 shows cases where the Chinese `A` gets translated into a prepositional phrase that expresses location.

   When `A` becomes a gerund phrase and an object of a preposition, it is also categorized in the `B` *preposition* `A` category (Example 2.3).

3. `A` 's `B`

   In this class, the English translation is an explicit *s*-genitive case, as in Example 3.1. This class occurs much less often, but is still interesting because of the difference from the *of*-genitive.

4. *relative clause*

   We include the obvious relative clause cases like Example 4.1 where a relative clause is introduced by a relative pronoun. We also include reduced relative clauses like Example 4.2 in this class.

5. `A` *preposition* `B`

   This class is another small one. The English translations that fall into this class usually have some number, percentage or level word in the Chinese `A`.

Some NPs are translated into a hybrid of these categories, or don't fit into one of the five categories, for instance, involving an adjectival pre-modifier and a relative clause. In those cases, they are put into an "other" category.[2]

5.2.2 Data annotation of DE classes

In order to train a classifier and test its performance, we use the Chinese Treebank 6.0 (LDC2007T36) and the English Chinese Translation Treebank 1.0 (LDC2007T02). The word alignment data (LDC2006E93) is also used to align the English and Chinese words between LDC2007T36 and LDC2007T02. The overlapping part of the three datasets are a subset of CTB6 files 1 to 325. After preprocessing those three sets of data, we have 3253 pairs of Chinese sentences and their translations. In those sentences, we use the gold-standard Chinese tree structure to get 3412 Chinese DEs in noun phrases that we want to annotate. Among the 3412 DEs, 530 of them are in the "other" category and are not used in the classifier training and evaluation. The statistics of the five classes are:

1. `A B`: 693 (24.05%)
2. `B` *preposition* `A`: 1381 (47.92%)
3. `A` 's `B`: 91 (3.16%)
4. *relative clause*: 669 (23.21%)

[2]The "other" category contains many mixed cases that could be difficult Chinese patterns to translate. We will leave this for future work.

1.	A B
1.1.	优越(*excellent*)/的(*DE*)/地理(*geographical*)/条件(*qualification*) → "excellent geographical qualifications"
1.2.	我们(*our*)/的(*DE*)/金融(*financial*)/风险(*risks*) → "our financial risks"
1.3.	贸易(*trade*)/的(*DE*)/互补性(*complement*) → "trade complement"
2.	B ***preposition*** A
2.1.	投资(*investment*)/环境(*environment*)/的(*DE*)/改善(*improvement*) → "the improvement of the investment environment"
2.2.	崇明县(*Chongming county*)/内(*inside*)/的(*DE*)/单位(*organization*) → "organizations inside Chongming county"
2.3.	一(*one*)/个(*measure word*)/观察(*observe*)/中国(*China*)/市场(*market*)/的(*DE*)/小小(*small*)/窗口(*window*) → "a small window for watching over Chinese markets"
3.	A **'s** B
3.1.	国家(*nation*)/的(*DE*)/宏观(*macro*)/管理(*management*) → "the nation 's macro management"
4.	***relative clause***
4.1.	中国(*China*)/不能(*cannot*)/生产(produce)/而(*and*)/又(*but*)/很(*very*)/需要(*need*)/的(*DE*)/药品(*medicine*) → "medicine that cannot be produced by China but is urgently needed"
4.2.	外商(*foreign business*)/投资(*invest*)/企业(*enterprise*)/获得(*acquire*)/的(*DE*)/人民币(*RMB*)/贷款(*loan*) → "the loans in RMB acquired by foreign-invested enterprises"
5.	A ***preposition*** B
5.1.	四千多万(*more than 40 million*)/美元(*US dollar*)/的(*DE*)/产品(*product*) → more than 40 million US dollars in products

Table 5.1: Examples for the 5 DE classes

5. A *preposition* B: 48 (1.66%)

The way we annotated the 3412 Chinese DEs in noun phrases is semi-automatic. Since we have the word alignment data (LDC2006E93) and the Chinese parse trees, we wrote programs to check the Chinese NP boundary, and find the corresponding English translation from the word alignment. Then the program checks where the character 的 is aligned to, and checks whether the Chinese words around 的 are reordered or kept in the same order. In some cases it's very clear. For example, if 的 is aligned to a preposition (e.g., "with"), all the Chinese words in front of 的 are aligned to English words after "with", and all the Chinese words behind 的 are aligned to English in front of "with", then the program automatically annotate this case as B *preposition* A. About half of the examples we

annotated were covered by the rules, so we only had to manually annotated the rest.

It is possible to do annotations without the parse trees and/or the word alignment data. But then an annotator will have to check for every 的, (*i*) whether it's part of an NP, (*ii*) mark the range of the Chinese NP, (*iii*) identify the corresponding translation in English, (*iv*) determine which of the 5 class (or other) this 的 in NP belongs to.

5.2.3 Discussion on the "other" class

In addition to the five classes, some DEs in NPs fall into the "other" class. The "other" class contains more complicated examples like when the NP gets translated into discontinuous fragments, or when the B part in "A 的 B" gets translated to a verb phrase, etc. For example, the NP with DE "中国 经济 的 不断 发展" was translated into "China 's economy has continued to develop" in one sentence. Another example of the "other" class is apposition. We see examples like "本 届 锦标赛 女子 个人 全能 银牌 得主 罗马尼亚 的 米洛索维奇" translates into "Romanian Mirosoviki , silver medal winner for the overall championships". The part before 的 was translated into an apposition in English, and the word orders were reversed around 的. This could potentially be separated out as another class, but in our experiments we decided it's a small class and marked it as "other".

5.3 Log-linear DE classifier

In order to see how well we can categorize DEs in noun phrases into one of the five classes, we train a log-linear classifier to classify each DE according to features extracted from its surrounding context. Since we want the training and testing conditions to match, when we extract features for the classifier, we don't use gold-standard parses. Instead, we use a parser trained on CTB6 excluding files 1-325. We then use this parser to parse the 3253 Chinese sentences with the DE annotation and extract parse-related features from there.

5.3.1 Experimental setting

For the classification experiment, we exclude the "other" class and only use the 2882 examples that fall into the five pre-defined classes. To evaluate the classification performance

	5-class Acc. (%)	2-class Acc. (%)
majority	47.9	71.1
baseline	-	76.0
DEPOS	54.8	71.0
+A-pattern	67.9	83.7
+POS-ngram	72.1	84.9
+Lexical	74.9	86.5
+SemClass	75.1	86.7
+Topicality	75.4	86.9

Table 5.2: 5-class and 2-class classification accuracy. "baseline" is the heuristic rules in (Wang et al., 2007). "majority" is labeling everything as the largest class. Others are various features added to the log-linear classifier.

and understand what features are useful, we compute the accuracy by averaging five 10-fold cross-validations.[3]

As a baseline, we use the rules introduced in Wang et al. (2007) to decide if the DEs require reordering or not. However, since their rules only decide if there is reordering in an NP with DE, their classification result only has two classes. In order to compare our classifier's performance with the rules in Wang et al. (2007), we have to map our five-class results into two classes. So we mapped *B preposition A* and *relative clause* into the class "*reordered*", and the other three classes into "*not-reordered*".

5.3.2 Feature Engineering

To understand which features are useful for DE classification, we list our feature engineering steps and results in Table 5.2. In Table 5.2, the 5-class accuracy is defined by:

$$\frac{\text{(number of correctly labeled DEs)}}{\text{(number of all DEs)}} \times 100$$

The 2-class accuracy is defined similarly, but it is evaluated on the 2-class "*reordered*" and "*not-reordered*" after mapping from the 5 classes.

The DEs we are classifying are within an NP. We refer to them as $[\text{A 的 B}]_{\text{NP}}$. A

[3]We evaluate the classifier performance using cross-validations to get the best setting for the classifier. The proof of efficacy of the DE classifier is MT performance on independent data in Section 5.5.

includes all the words in the NP before 的; B includes all the words in the NP after 的. To illustrate, we will use the following NP:

[[韩国 最 大]$_A$ 的 [投资 对象国]$_B$]$_{NP}$

Korea most big DE investment target country

Translation: Korea's largest target country for investment

to show examples of each feature. The parse structure of the NP is listed in Figure 5.2.

```
(NP
  (NP (NR 韩国))
  (CP
    (IP
      (VP
        (ADVP (AD 最))
        (VP (VA 大))))
    (DEC 的))
  (NP (NN 投资) (NN 对象国))))))
```

Figure 5.2: The parse tree of the Chinese NP.

DEPOS: part-of-speech tag of DE

Since the part-of-speech tag of DE indicates its syntactic function, it is the first obvious feature to add. The NP in Figure 5.2 will have the feature "DEC". This basic feature will be referred to as DEPOS. Note that since we are only classifying DEs in NPs, ideally the part-of-speech tag of DE will either be DEC or DEG as described in Section 5.2. However, since we are using automatic parses instead of gold-standard ones, the DEPOS feature might have other values than just DEC and DEG. From Table 5.2, we can see that with this simple feature, the 5-class accuracy is low, but is at least better than simply guessing the majority class (47.92%). The 2-class accuracy is still lower than using the heuristic rules in (Wang et al., 2007), which is reasonable because their rules encode more information than just the POS tags of DEs.

A-pattern: Chinese syntactic patterns appearing before 的

Secondly, we want to incorporate the rules in (Wang et al., 2007) as features in the log-linear classifier. We added features for certain indicative patterns in the parse tree (listed in Table 5.3).

1. **A is ADJP**: true if A+DE is a DNP, which is in the form of "ADJP+DEG".
2. **A is QP**: true if A+DE is a DNP, which is in the form of "QP+DEG".
3. **A is pronoun**: true if A+DE is a DNP, which is in the form of "NP+DEG", and the NP is a pronoun.
4. **A ends with VA**: true if A+DE is a CP, which is in the form of "IP+DEC", and the IP ends with a VP that's either just a VA or a VP preceded by a ADVP.

Table 5.3: A-pattern features

Features 1–3 are inspired by the rules in (Wang et al., 2007), and the fourth rule is based on the observation that even though the predicative adjective VA acts as a verb, it actually corresponds to adjectives in English as described in (Xia, 2000).[4] We call these four features A-pattern. Our example NP in Figure 5.2 will have the fourth feature "A ends with VA" in Table 5.3, but not the other three features. In Table 5.2 we can see that after adding A-pattern, the 2-class accuracy is already much higher than the baseline. We attribute this to the fourth rule and also to the fact that the classifier can learn weights for each feature.[5] Indeed, not having a special case for VA stative verbs is a significant oversight in the rules of (Wang et al., 2007).

POS-ngram: unigrams and bigrams of POS tags

The POS-ngram feature adds all unigrams and bigrams in A and B. Since A and B have different influences on the choice of DE class, we distinguish their ngrams into two sets of features. We also include the bigram pair across DE which gets another feature name

[4]Quote from (Xia, 2000): "VA roughly corresponds to adjectives in English and stative verbs in the literature on Chinese grammar."

[5]We also tried extending a rule-based 2-class classifier with the fourth rule. The accuracy is 83.48%, only slightly lower than using the same features in a log-linear classifier.

for itself. The example NP in Figure 5.2 will have these features (we use b to indicate boundaries):

- POS unigrams in A: "NR", "AD", "VA"
- POS bigrams in A: "b-NR", "NR-AD", "AD-VA", "VA-b"
- cross-DE POS bigram: "VA-NN"
- POS unigram in B: "NN"
- POS bigrams in B: "b-NN", "NN-NN", "NN-b"

The part-of-speech ngram features add 4.24% accuracy to the 5-class classifier.

Lexical: lexical features

In addition to part-of-speech features, we also tried to use features from the words themselves. But since using full word identity resulted in a sparsity issue,[6] we take the one-character suffix of each word and extract suffix unigram and bigram features from them. The argument for using suffixes is that it often captures the larger category of the word (Tseng et al., 2005). For example, 中国 (China) and 韩国 (Korea) share the same suffix 国, which means "country". These suffix ngram features will result in these features for the NP in Figure 5.2:

- suffix unigrams: "国", "最", "大", "的", "资", "国"
- suffix bigrams: "b-国", "国-最", "最-大", "大-的", "的-资", "资-国", "国-b"

Other than the suffix ngram, we also add three other lexical features: first, if the word before DE is a noun, we add a feature that is the conjunction of POS and suffix unigram. Secondly, an "NR only" feature will fire when `A` only consists of one or more NRs (that is, proper nouns). Thirdly, we normalize different forms of "percentage" representation, and add a feature if they exist. This includes words that start with "百分之" or end with the percentage sign "%". The first two features are inspired by the fact that a noun and its

[6] The accuracy is worse when we tried using the word identity instead of the suffix.

type can help decide "B prep A" versus "A B". Here we use the suffix of the noun and the NR tag to help capture its animacy, which is useful in choosing between the *s*-genitive (*the boy's mother*) and the *of*-genitive (*the mother of the boy*) in English (Rosenbach, 2003). The third feature is added because many of the cases in the "A *preposition* B" class have a percentage number in A. We call these sets of features Lexical. Together they provide a 2.73% accuracy improvement over the previous setting.

SemClass: semantic class of words

We also use a Chinese thesaurus, CiLin, to look up the semantic classes of the words in [A 的 B] and use them as features. CiLin is a Chinese thesaurus published in 1984 (Mei et al., 1984). CiLin is organized in a conceptual hierarchy with five levels. We use the level-1 tags, which include 12 categories.[7] This feature fires when a word we look up has one level-1 tag in CiLin. This kind of feature is referred to as SemClass in Table 5.2. For the example in Figure 5.2, two words have a single level-1 tag: "最"(most) has a level-1 tag K[8] and "投资"(investment) has a level-1 tag H[9]. "韩国" and "对象国" are not listed in CiLin, and "大" has multiple entries. Therefore, the SemClass features are: (*i*) before DE: "K"; (*ii*) after DE: "H"

Topicality: re-occurrence of nouns

The last feature we add is a Topicality feature, which is also useful for disambiguating *s*-genitive and *of*-genitive. We approximate the feature by caching the nouns in the previous two sentences, and fire a topicality feature when the noun appears in the cache. Take this NP in MT06 as an example:

"南韩 与 北韩 的 奥运 代表队"

For this NP, all words before DE and after DE appeared in the previous sentence. Therefore the topicality features "cache-before-DE" and "cache-after-DE" both fire.

[7] We also tried adding more levels but it did not help.
[8] K is the category 助语 (auxiliary) in CiLin.
[9] H is the category 活动 (activities) in CiLin.

After all the feature engineering above, the best accuracy on the 5-class classifier we have is 75.4%, which maps into a 2-class accuracy of 86.9%. Comparing the 2-class accuracy to the (Wang et al., 2007) baseline, we have a 10.9% absolute improvement. The 5-class accuracy and confusion matrix is listed in Table 5.4.

real →	A 's B	AB	A *prep.* B	B *prep.* A	*rel. clause*
A 's B	168	36	0	110	0
AB	48	2473	73	227	216
A *prep.* B	0	18	46	23	11
B *prep.* A	239	691	95	5915	852
rel. clause	0	247	26	630	2266
Total	455	3465	240	6905	3345
Accuracy(%)	36.92	71.37	19.17	85.66	67.74

Table 5.4: The confusion matrix for 5-class DE classification

"A *preposition* B" is a small category and is the most confusing. "A 's B" also has lower accuracy, and is mostly confused with "B *preposition* A". This could be due to the fact that there are some cases where the translation is correct both ways, but also could be because the features we added have not captured the difference well enough.

5.4 Labeling and Reordering "DE" Constructions

The DE classifier uses Chinese CFG parses generated from the Stanford Chinese Parser (Levy and Manning, 2003). The parses are used to design features for the DE classifier, as well as performing reordering on the Chinese trees so the word orders can match better with English. In this section we will look at the DE constructions in the automatically parsed MT training data (which contain errors) and explain which DEs get labeled with the five classes. We will also explain in more detail how we perform reordering once the DEs are labeled.

There are 476,007 的s as individual words in the MT training data (described in Section 5.5.1. The distribution of their POS tags is in Table 5.5. In this distribution, we see that 的s get tags other than the ones mentioned in the guidelines (DEC, DEG, DEV, SP, DER, or AS). We do not mark all the 的s, but only mark the 的s under NPs with the POS tags

POS of 的	counts
DEG	250458
DEC	219969
DEV	3039
SP	1677
DER	263
CD	162
X	156
PU	79
NN	65
AS	64
ON	45
FW	11
AD	6
M	3
NR	3
P	3
NT	2
CC	1
VV	1

Table 5.5: The distribution of the part-of-speech tags of DEs in the MT training data.

DEC or DEG, as shown in Figure 5.3. There are 461,951 的s that are processed in the MT training data; the remaining 14,056 are unlabeled. More details are in Table 5.7.

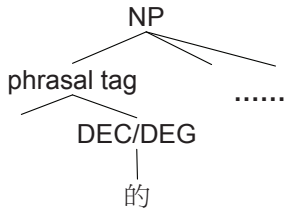


Figure 5.3: The NP with DEs.

After we label the 的s with one of the five classes, we also reorder the ones with the two classes $的_{relc}$ and $的_{BprepA}$. The way we reorder is similar to (Wang et al., 2007). For each NP with DEs, we move the pre-modifying modifiers with $的_{relc}$ or $的_{BprepA}$ to the position behind the NP. As an example, we choose an NP with DE from the MT training data. In

Figure 5.4 there is a CP (俄/Russia 美/US 总统/president 今年/this year 举行/hold 的/DE) and a QP (三/three 次/times) that pre-modifies the NP (会晤/meeting). The 的 (DE) is labeled as 的$_{\text{relc}}$; therefore it needs to be reordered. The right tree in Figure 5.4 shows that the CP gets reordered to post-modify the NP, and the 的$_{\text{relc}}$ also get reordered to be at the front of the CP. The other modifier, QP, remains pre-modifying the noun phrase 会晤/meeting.

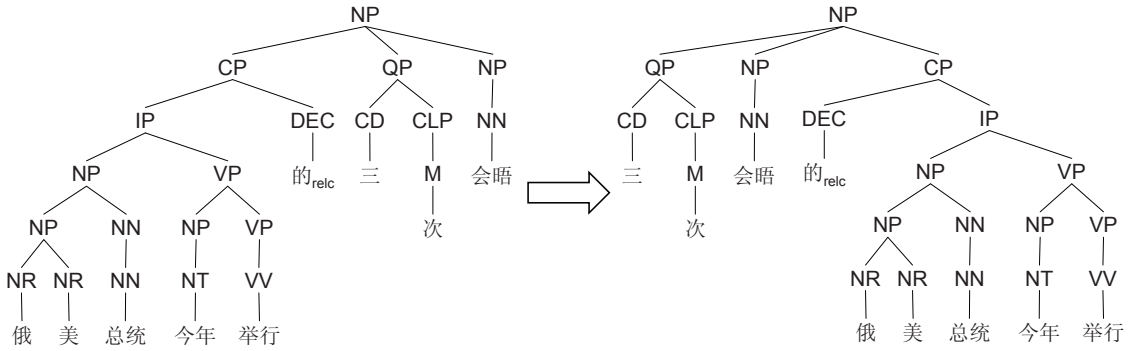


Figure 5.4: Reorder an NP with DE. Only the pre-modifier with DE (a CP in this example) is reordered. The other modifiers (a QP in this example) stay in the same place.

5.5 Machine Translation Experiments

5.5.1 Experimental Setting

For our MT experiments, we used *Phrasal*, a re-implementation of Moses (Koehn et al., 2003), a state-of-the-art phrase-based system. The alignment is done by the Berkeley word aligner (Liang et al., 2006) and then we symmetrized the word alignment using the grow-diag heuristic. For features, we incorporate Moses' standard eight features as well as the lexicalized reordering model. Parameter tuning is done with Minimum Error Rate Training (MERT) (Och, 2003). The tuning set for MERT is the NIST MT06 data set, which includes 1664 sentences. We evaluate the result with MT02 (878 sentences), MT03 (919 sentences), and MT05 (1082 sentences).

Our MT training corpus contains 1,560,071 sentence pairs from various parallel corpora

from LDC.[10] There are 12,259,997 words on the English side. Chinese word segmentation is done by the Stanford Chinese segmenter (Chang et al., 2008). After segmentation, there are 11,061,792 words on the Chinese side. We use a 5-gram language model trained on the Xinhua and AFP sections of the Gigaword corpus (LDC2007T40) and also the English side of all the LDC parallel data permissible under the NIST08 rules. Documents of Gigaword released during the epochs of MT02, MT03, MT05, and MT06 were removed.

To run the DE classifier, we also need to parse the Chinese texts. We use the Stanford Chinese parser (Levy and Manning, 2003) to parse the Chinese side of the MT training data and the tuning and test sets.

5.5.2 Baseline Experiments

We have two different settings as baseline experiments. The first is without reordering or DE annotation on the Chinese side; we simply align the parallel texts, extract phrases and tune parameters. This experiment is referred to as BASELINE. Also, we reorder the training data, the tuning, and the test sets with the NP rules in (Wang et al., 2007) and compare our results with this second baseline (WANG-NP).

The NP reordering preprocessing (WANG-NP) showed consistent improvement in Table 5.6 on all test sets, with BLEU point gains ranging from 0.15 to 0.40. This confirms that having reordering around DEs in NP helps Chinese-English MT.

5.5.3 Experiments with 5-class DE annotation

We use the best setting of the DE classifier described in Section 5.3 to annotate DEs in NPs in the MT training data as well as the NIST tuning and test sets.[11] If a DE is in an NP, we use the annotation of 的$_{\text{AB}}$, 的$_{\text{AsB}}$, 的$_{\text{BprepA}}$, 的$_{\text{relc}}$, or 的$_{\text{AprepB}}$ to replace the original DE character. Once we have the DEs labeled, we preprocess the Chinese sentences by

[10] LDC2002E18, LDC2003E07, LDC2003E14, LDC2005E83, LDC2005T06, LDC2006E26, LDC2006E85, LDC2002L27, and LDC2005T34.

[11] The DE classifier used to annotate the MT experiment was trained on all the available data described in Section 5.2.2.

BLEU				
	MT06(tune)	MT02	MT03	MT05
BASELINE	32.39	32.51	32.75	31.42
WANG-NP	32.75(+0.36)	32.66(+0.15)	33.15(+0.40)	31.68(+0.26)
DE-Annotated	**33.39**(+1.00)	**33.75**(+1.24)	**33.63**(+0.88)	**32.91**(+1.49)
BASELINE+Hier	32.96	33.10	32.93	32.23
DE-Annotated+Hier	**33.96**(+1.00)	**34.33**(+1.23)	**33.88**(+0.95)	**33.01**(+0.77)
Translation Error Rate (TER)				
	MT06(tune)	MT02	MT03	MT05
BASELINE	61.10	63.11	62.09	64.06
WANG-NP	59.78(−1.32)	62.58(−0.53)	61.36(−0.73)	62.35(−1.71)
DE-Annotated	**58.21**(−2.89)	**61.17**(−1.94)	**60.27**(−1.82)	**60.78**(−3.28)

Table 5.6: MT experiments with different settings on various NIST MT evaluation datasets. We used both the BLEU and TER metrics for evaluation. All differences between DE-Annotated and BASELINE are significant at the level of 0.05 with the approximate randomization test in (Riezler and Maxwell, 2005).

reordering them.[12] Note that not all DEs in the Chinese data are in NPs, therefore not all DEs are annotated with the extra labels. Table 5.7 lists the statistics of the DE classes in the MT training data.

class of 的 (DE)	counts	percentage
$的_{AB}$	112,099	23.55%
$的_{AprepB}$	2,426	0.51%
$的_{AsB}$	3,430	0.72%
$的_{BprepA}$	248,862	52.28%
$的_{relc}$	95,134	19.99%
的 (unlabeled)	14,056	2.95%
total number of 的	476,007	100%

Table 5.7: The number of different DE classes labeled for the MT training data.

After this preprocessing, we restart the whole MT pipeline – align the preprocessed data, extract phrases, run MERT and evaluate. This setting is referred to as DE-Annotated in Table 5.6.

[12] Reordering is applied on DNP and CP for reasons described in Wang et al. (2007). We reorder only when the 的 is labeled as $的_{BprepA}$ or $的_{relc}$.

class	MT02	MT03	MT05
的$_{\text{AB}}$	178	201	261
的$_{\text{AprepB}}$	9	8	5
的$_{\text{AsB}}$	9	11	13
的$_{\text{BprepA}}$	544	517	671
的$_{\text{relc}}$	271	296	331
的 (unlabeled)	29	44	63
Reordered DE(%)	80.6	78.6	78.2

Table 5.8: Counts of each 的 and its labeled class in the three test sets.

5.5.4 Hierarchical Phrase Reordering Model

To demonstrate that the technique presented here is effective even with a hierarchical decoder, we conducted additional experiments with a hierarchical phrase reordering model introduced by Galley and Manning (2008). The hierarchical phrase reordering model can handle the key examples often used to motivate syntax-based systems; therefore we think it is valuable to see if the DE annotation can still improve on top of that. In Table 5.6, BASELINE+Hier gives consistent BLEU improvement over BASELINE. Using DE annotation on top of the hierarchical phrase reordering models (DE-Annotated+Hier) provides extra gain over BASELINE+Hier. This shows the DE annotation can help a hierarchical system. We think similar improvements are likely to occur with other hierarchical systems.

5.6 Analysis

5.6.1 Statistics on the Preprocessed Data

Since our approach DE-Annotated and one of the baselines (WANG-NP) are both preprocessing Chinese sentences, knowing what percentage of the sentences are altered will be one useful indicator of how different the systems are from the baseline. In our test sets, MT02 has 591 out of 878 sentences (67.3%) that have DEs under NPs; for MT03 it is 619 out of 919 sentences (67.4%); for MT05 it is 746 out of 1082 sentences (68.9%). This shows that our preprocessing affects the majority of the sentences and thus it is not surprising that preprocessing based on the DE construction can make a significant difference. We provide more detailed counts for each class in all the test sets in Table 5.8.

5.6.2 Example: how DE annotation affects translation

Our approach DE-Annotated reorders the Chinese sentence, which is similar to the approach proposed by Wang et al. (2007) (WANG-NP). However, our focus is on the annotation on DEs and how this can improve translation quality. Table 5.9 shows an example that contains a DE construction that translates into a relative clause in English.[13] The automatic parse tree of the sentence is listed in Figure 5.5. The reordered sentences of WANG-NP and DE-Annotated appear on the top and bottom in Figure 5.6. For this example, both systems decide to reorder, but DE-Annotated had the extra information that this 的 is a 的$_{\text{relc}}$. In Figure 5.6 we can see that in WANG-NP, "的" is being translated as "for", and the translation afterwards is not grammatically correct. On the other hand, the bottom of Figure 5.6 shows that with the DE-Annotated preprocessing, "的$_{\text{relc}}$" is now translated into "which was" and well connected with the later translation. This shows that disambiguating 的 (DE) helps in choosing a better English translation.

Chinese	比亚吉 曾 协助 草拟 [一 份 遭 工会 和 左翼 分子 强烈 反对]$_{\text{A}}$ 的 [就业 改革 方案]$_{\text{B}}$ 。
Ref 1	biagi had assisted in drafting [an employment reform plan]$_{\text{B}}$ [that was strongly opposed by the labor union and the leftists]$_{\text{A}}$.
Ref 2	biagi had helped in drafting [a labor reform proposal]$_{\text{B}}$ [that provoked strong protests from labor unions and the leftists]$_{\text{A}}$.
Ref 3	biagi once helped drafting [an employment reform scheme]$_{\text{B}}$ [that was been strongly opposed by the trade unions and the left - wing]$_{\text{A}}$.
Ref 4	biagi used to assisted to draft [an employment reform plan]$_{\text{B}}$ [which is violently opposed by the trade union and leftest]$_{\text{A}}$.

Table 5.9: A Chinese example from MT02 that contains a DE construction that translates into a relative clause in English. The []$_{\text{A}}$ []$_{\text{B}}$ is hand-labeled to indicate the approximate translation alignment between the Chinese sentence and English references.

[13] In this example, all four references agreed on the relative clause translation. Sometimes DE constructions have multiple appropriate translations, which is one of the reasons why certain classes are more confusable in Table 5.4.

```
(IP
 (NP (NN 比亚吉))
 (VP
   (ADVP (AD 曾))
   (VP (VV 协助)
     (IP
       (VP (VV 草拟)
         (NP
           (QP (CD 一)
             (CLP (M 份)))
           (CP
             (IP
               (VP (VV 遭)
                 (NP
                   (NP (NN 工会)
                     (CC 和)
                     (NN 左翼) (NN 分子))
                   (ADJP (JJ 强烈))
                   (NP (NN 反对)))))
             (DEC 的))
           (NP (NN 就业) (NN 改革) (NN 方案)))))))
 (PU 。))
```

Figure 5.5: The parse tree of the Chinese sentence in Table 5.9.

5.7 Conclusion

In this chapter, we presented a classification of Chinese 的 (DE) constructions in NPs according to how they are translated into English. We applied this DE classifier to the Chinese sentences of MT data, and we also reordered the constructions that required reordering to better match their English translations. The MT experiments showed our preprocessing gave significant BLEU and TER score gains over the baselines. Based on our classification and MT experiments, we found that not only do we have better rules for deciding what to reorder, but the syntactic, semantic, and discourse information that we capture in the Chinese sentence allows us to give hints to the MT system, which allows better translations to be chosen.

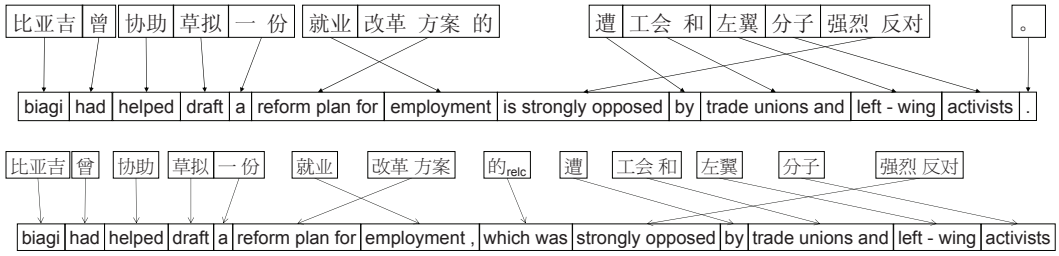


Figure 5.6: The top translation is from WANG-NP of the Chinese sentence in Table 5.9. The bottom one is from DE-Annotated. In this example, both systems reordered the NP, but DE-Annotated has an annotation on the 的 (DE).

The DE classifier preprocessing approach can also be used for other syntax-based systems directly. For systems that need only the sentences from the source language, the reordered and labeled sentence can be the input, which could provide further help in longer-distance reordering cases. For systems that use the parse tress from the source language, the DE preprocessing approach can provide an reordered parsed tree (e.g., Figure 5.4). Also, since our log-linear model can generate probability distributions of each classes, another possibility is not to preprocess and generate one reordered sentence, but to generate a lattice with possible reordering and labeling choices, and run the decoder on the lattice instead of on one preprocessed sentence.

Also, there are more function words in Chinese that can cause reordering. For example, prepositions and localizers (postpositions), or 把(BA) and 被(BEI) constructions are all likely to be reordered when translating into English. The technique of disambiguating and reordering beforehand in this chapter can be extended for more cases.

Chapter 6

Conclusions and Future Work

The focus of this dissertation is the investigation of important differences between the Chinese and English languages, and how they have made machine translation from Chinese to English more difficult.

We carefully studied state-of-the-art MT system outputs, and identified linguistic issues that currently pose obstacles for Chinese-English MT. We found that the Chinese language is difficult from bottom up – starting from its writing system, morphology, up to the syntactic structures, and all the way to discourse structures.

The Chinese writing system does not have explicit word boundaries in between words, therefore word segmentation is an essential first step for Chinese NLP tasks. We found that a general definition of "word" is not necessarily the best fit for MT systems, but instead, we found that tuning the Chinese (source) word length for a granularity better matched with English (target) words works well. We also found that increasing the consistency of the segmentation is very important for MT, which we achieved by integrating lexicon-based features to a feature-rich conditional random field segmenter. We also found that jointly identifying proper nouns with segmentation improved segmentation quality for MT as well. We think integrating more source-side word information, such as named entities, and using this information more tightly inside the MT system will further improve the system performance and understandability of the output.

One level up from word-level ambiguity, we studied sentence-level source-side information. Chinese and English are both SVO (subject-verb-object) languages, so there are

some similar syntactic constructions between them. But there also exist distinctive word ordering differences, for example different phrasal ordering of prepositional phrases and verb phrases, or constructions that are specific to Chinese. To fully describe the Chinese syntactic structures and utilize them in a MT system, we designed a set of Chinese grammatical relations following the design principles of the Stanford English typed dependencies. Our Chinese grammatical relation representation has the advantages of high coverage, good readability and being descriptive of relations that are similar to English as well as relations that are specific to Chinese. By using the grammatical relations, we showed improvement in discriminative word reordering models that can be easily applied to a phrase-based MT system. In terms of MT, we think our grammatical relations can provide even more source-side syntactic information if they can be directly integrated into a dependency-based decoding framework. We also think this set of grammatical relations should be useful for other Chinese NLP tasks, especially meaning extraction related tasks.

Since there are several ambiguous Chinese constructions, we also explored the possibility of disambiguating them earlier in the MT process. We focused on the most common function word "的", which does not have a direct translation into English and can oftentimes lead to ambiguous translations with longer-distance word reordering. According to our data analysis, we categorized the usage of DE into five most prominent classes, and labeled some data for training and developing a good DE classifier. In our experiments, we showed that we can build a classifier with good performance by using features with lexical, semantic, syntactic and discourse contexts. We use the DE classifier to explicitly mark the DE usages in the source data and reorder the cases where Chinese word orders differ from English. By doing this, we were able to show significant improvement on our MT experiments. This proved that analyzing and disambiguating function words in Chinese can lead to a big improvement that the built-in reordering models could not capture in the baseline systems. We think for future directions, it will be worthwhile to identify more Chinese specific constructions or function words that are likely to cause ambiguous translations or translations with longer distance word reordering, and then to build classifiers to further disambiguate them on the source side.

In addition to the issues we addressed in the dissertation, we also observed much higher level linguistic issues that were causing errors in the current state-of-the-art MT systems.

The current framework of this thesis is sentence-based translation, where all the processing and translation steps are done based on the assumption that each sentence can be translated independent of the surrounding context. According to our analysis, we found that Chinese sentences are likely to drop pronouns that have been mentioned in previous sentences, leaving a zero anaphora that refers to a noun that does not appear in the sentence to be translated. Since in English the pronouns are usually necessary for translation, it is not enough to just use the current sentence. Other problematic higher-level issues which require discourse context include choosing the right tense and aspect in the English translation and working out the correct way to link clauses in the translation so that they properly convey the intended discourse relations. Therefore we believe even higher level source-side information, such as discourse structures on the source side, will be essential for making Chinese-English MT better. Currently there hasn't been much work addressing the issue of discourse structures or even the within-sentence clause linkage problem, we think this will be an important area for future research on Chinese-English MT.

Bibliography

Al-Onaizan, Y. and K. Papineni (2006, July). Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, pp. 529–536. Association for Computational Linguistics.

Andrew, G. (2006, July). A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, pp. 465–472. Association for Computational Linguistics.

Avramidis, E. and P. Koehn (2008, June). Enriching morphologically poor languages for statistical machine translation. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, pp. 763–770. Association for Computational Linguistics.

Badr, I., R. Zbib, and J. Glass (2009, March). Syntactic phrase reordering for English-to-Arabic statistical machine translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, Athens, Greece, pp. 86–93. Association for Computational Linguistics.

Bender, E. (2000). The syntax of Mandarin bă: Reconsidering the verbal analysis. *Journal of East Asian Linguistics 9*, 105–145.

Birch, A., P. Blunsom, and M. Osborne (2009, March). A quantitative analysis of reordering phenomena. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, pp. 197–205. Association for Computational Linguistics.

Brown, P. F., V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics 19*(2), 263–311.

Carpuat, M. and D. Wu (2007, June). Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 61–72. Association for Computational Linguistics.

Chan, Y. S., H. T. Ng, and D. Chiang.

Chang, P.-C., M. Galley, and C. D. Manning (2008, June). Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, pp. 224–232. Association for Computational Linguistics.

Chang, P.-C., D. Jurafsky, and C. D. Manning (2009, March). Disambiguating "DE" for Chinese-English machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, pp. 215–223. Association for Computational Linguistics.

Chang, P.-C. and K. Toutanova (2007, June). A discriminative syntactic word order model for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 9–16. Association for Computational Linguistics.

Chang, P.-C., H. Tseng, D. Jurafsky, and C. D. Manning (2009, June). Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, Boulder, Colorado.

Chen, C.-Y., S.-F. Tseng, C.-R. Huang, and K.-J. Chen (1993). Some distributional properties of Mandarin Chinese–a study based on the Academia Sinica corpus.

Chiang, D. (2005, June). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, pp. 263–270. Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics 33*(2), 201–228.

Chodorow, M., J. Tetreault, and N.-R. Han (2007, June). Detection of grammatical errors involving prepositions. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, Prague, Czech Republic, pp. 25–30. Association for Computational Linguistics.

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

de Marneffe, M.-C., B. Maccartney, and C. D. Manning (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, pp. 449–454.

de Marneffe, M.-C. and C. D. Manning (2008, August). The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, Manchester, UK, pp. 1–8. Coling 2008 Organizing Committee.

Deng, Y. and W. Byrne (2005). HMM word and phrase alignment for statistical machine translation. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, pp. 169–176. Association for Computational Linguistics.

Dugast, L., J. Senellart, and P. Koehn (2007, June). Statistical post-editing on SYSTRAN's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, pp. 220–223. Association for Computational Linguistics.

Dyer, C. (2009, June). Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, pp. 406–414. Association for Computational Linguistics.

Dyer, C., S. Muresan, and P. Resnik (2008, June). Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, pp. 1012–1020. Association for Computational Linguistics.

Emerson, T. (2005). The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Finkel, J. R., T. Grenager, and C. Manning (2005, June). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, pp. 363–370. Association for Computational Linguistics.

Fraser, A. and D. Marcu (2007). Measuring word alignment quality for statistical machine translation. *Computational Linguistics 33*(3), 293–303.

Galley, M., J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer (2006, July). Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, pp. 961–968. Association for Computational Linguistics.

Galley, M., M. Hopkins, K. Knight, and D. Marcu. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*.

Galley, M. and C. D. Manning (2008, October). A simple and effective hierarchical phrase reordering model. In *Proceedings of EMNLP*, Honolulu, Hawaii, pp. 847–855. Association for Computational Linguistics.

Gao, J., M. Li, A. Wu, and C.-N. Huang (2005). Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*.

Gao, J., A. Wu, M. Li, C.-N. Huang, H. Li, X. Xia, and H. Qin (2004). Adaptive Chinese word segmentation. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, pp. 462. Association for Computational Linguistics.

Habash, N. and F. Sadat (2006, June). Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, New York City, USA, pp. 49–52. Association for Computational Linguistics.

Hollink, V., J. Kamps, C. Monz, and M. de Rijke (2004). Monolingual document retrieval for European languages. *Information Retrieval 7*(1).

Huang, F. and K. Papineni (2007, June). Hierarchical system combination for machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 277–286. Association for Computational Linguistics.

Huang, L. and D. Chiang (2007, June). Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 144–151. Association for Computational Linguistics.

Huang, L., K. Knight, and A. Joshi (2006). Statistical syntax-directed translation with extended domain of locality. In *Proceedings of AMTA*, Boston, MA.

Ittycheriah, A. and S. Roukos (2007, April). Direct translation model 2. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, Rochester, New York, pp. 57–64. Association for Computational Linguistics.

Koehn, P., H. Hoang, A. B. Mayne, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst (2007).

Moses: Open source toolkit for statistical machine translation. In *ACL, Demonstration Session*.

Koehn, P. and K. Knight (2003). Empirical methods for compound splitting. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pp. 187–193. Association for Computational Linguistics.

Koehn, P., F. J. Och, and D. Marcu (2003). Statistical phrase-based translation. In *Proc. of NAACL-HLT*.

Lafferty, J., A. McCallum, and F. Pereira (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*.

Lee, J. and S. Seneff (2008, June). Correcting misuse of verb forms. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, pp. 174–182. Association for Computational Linguistics.

Lee, Y.-S., Y. Al-Onaizan, K. Papineni, and S. Roukos (2006, June). IBM spoken language translation system. In *TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, pp. 13–18.

Lee, Y.-S. and S. Roukos (2004). IBM spoken language translation system evaluation. In *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, pp. 39–46.

Levow, G.-A. (2006, July). The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proc. of the Fifth SIGHAN Workshop on Chinese Language Processing*.

Levy, R. and C. Manning (2003). Is it harder to parse Chinese, or the Chinese treebank? In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, pp. 439–446. Association for Computational Linguistics.

Lewis, II, P. M. and R. E. Stearns (1968). Syntax-directed transduction. *J. ACM 15*(3), 465–488.

Li, C. N. and S. A. Thompson (1976). *Subject and Topic: A New Typology of Language*. New York: Academic Press.

Liang, P., B. Taskar, and D. Klein (2006, June). Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, New York City, USA, pp. 104–111. Association for Computational Linguistics.

Marcu, D., W. Wang, A. Echihabi, and K. Knight (2006, July). SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, pp. 44–52. Association for Computational Linguistics.

Matusov, E., N. Ueffing, and H. Ney (2006, April). Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, pp. 33–40.

Mei, J.-j., Y.-M. Zheng, Y.-Q. Gao, and H.-X. Yin (1984). *TongYiCi CiLin*. Shanghai: the Commercial Press.

Minkov, E., R. Wang, A. Tomasic, and W. Cohen (2006, June). NER systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction. In *Proc. of NAACL-HLT, Companion Volume: Short Papers*, New York City, USA.

Ng, H. T. and J. K. Low (2004). Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based? In *Proc. of EMNLP*.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In E. Hinrichs and D. Roth (Eds.), *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 160–167.

Och, F. J., D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev (2004). A smorgasbord of features for statistical machine translation. In *Proceedings of HLT-NAACL*.

Och, F. J. and H. Ney (2002). Discriminative training and maximum entropy models for statistical machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, pp. 295–302. Association for Computational Linguistics.

Och, F. J. and H. Ney (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics 29*(1).

Och, F. J., C. Tillmann, and H. Ney (1999). Improved alignment models for statistical machine translation. In *Proc. of EMNLP*.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu (2001). BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Peng, F., F. Feng, and A. McCallum (2004). Chinese segmentation and new word detection using conditional random fields. In *Proc. of COLING*.

Peng, F., X. Huang, D. Schuurmans, and N. Cercone (2002). Investigating the relationship between word segmentation performance and retrieval performance in Chinese IR. In *Proc. of the 19th International Conference on Computational Linguistics*.

Quirk, C., A. Menezes, and C. Cherry (2005). Dependency treelet translation: Syntactically informed phrasal SMT. In *ACL*.

Riezler, S. and J. T. Maxwell (2005, June). On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan, pp. 57–64. Association for Computational Linguistics.

Rosenbach, A. (2003). Aspects of iconicity and economy in the choice between the s-genitive and the of-genitive in English. *Topics in English Linguistics 43*, 379–412.

Rosti, A.-V., S. Matsoukas, and R. Schwartz (2007, June). Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 312–319. Association for Computational Linguistics.

Shen, L., J. Xu, and R. Weischedel (2008, June). A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, pp. 577–585. Association for Computational Linguistics.

Shi, Y. and M. Wang (2007). A dual-layer CRFs based joint decoding method for cascaded segmentation and labeling tasks. In *IJCAI*.

Sproat, R. and T. Emerson (2003). The first international Chinese word segmentation bakeoff. In *Proceedings of the second SIGHAN workshop on Chinese language processing*, Morristown, NJ, USA, pp. 133–143. Association for Computational Linguistics.

Tillman, C. (2004). A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pp. 101–104.

Tillmann, C. (2003). A projection extension algorithm for statistical machine translation. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, Morristown, NJ, USA, pp. 1–8. Association for Computational Linguistics.

Tillmann, C. and F. Xia (2003). A phrase-based unigram model for statistical machine translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Morristown, NJ, USA, pp. 106–108. Association for Computational Linguistics.

Toutanova, K., H. Suzuki, and A. Ruopp (2008, June). Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, pp. 514–522. Association for Computational Linguistics.

Tseng, H., P. Chang, G. Andrew, D. Jurafsky, and C. Manning (2005). A conditional random field word segmenter for Sighan bakeoff 2005. In *Proc. of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Tseng, H., D. Jurafsky, and C. D. Manning (2005). Morphological features help POS tagging of unknown words across language varieties. In *Proc. of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Turner, J. and E. Charniak (2007, April). Language modeling for determiner selection. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, Rochester, New York, pp. 177–180. Association for Computational Linguistics.

Wang, C., M. Collins, and P. Koehn (2007, June). Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 737–745. Association for Computational Linguistics.

Wong, P.-k. and C. Chan (1996). Chinese word segmentation based on maximum matching and word binding force. In *Proceedings of the 16th conference on Computational linguistics*, Morristown, NJ, USA, pp. 200–203. Association for Computational Linguistics.

Wu, A. (2003). Customizable segmentation of morphologically derived words in Chinese. *International Journal of Computational Linguistics and Chinese Language Processing*.

Xia, F. (2000). The part-of-speech tagging guidelines for the Penn Chinese Treebank (3.0).

Xia, F. and M. McCord (2004, Aug 23–Aug 27). Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of Coling 2004*, Geneva, Switzerland, pp. 508–514. COLING.

Xu, J., R. Zens, and H. Ney (2004). Do we need Chinese word segmentation for statistical machine translation. In *Proc. of the Third SIGHAN Workshop on Chinese Language Learning*.

Xu, P., J. Kang, M. Ringgaard, and F. Och (2009, June). Using a dependency parser to improve SMT for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, pp. 245–253. Association for Computational Linguistics.

Xue, N. and L. Shen (2003). Chinese word segmentation as LMR tagging. In *Proceedings of the second SIGHAN workshop on Chinese language processing*, Morristown, NJ, USA, pp. 176–179. Association for Computational Linguistics.

Xue, N., F. Xia, F.-D. Chiou, and M. Palmer (2005). Building a large annotated Chinese corpus: the Penn Chinese treebank. *Journal of Natural Language Engineering 11*(2).

Yip, P.-C. and D. Rimmington (1997). *Chinese: An Essential Grammar*. Routledge.

Zens, R. and H. Ney (2006, June). Discriminative reordering models for statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, New York City, pp. 55–63. Association for Computational Linguistics.

Zhang, Y., R. Zens, and H. Ney (2007, April). Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, Rochester, New York, pp. 1–8. Association for Computational Linguistics.