Reordering in Statistical Machine Translation: A Function Word, Syntax-based Approach

Hendra Setiawan

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the School of Computing

NATIONAL UNIVERSITY OF SINGAPORE

2008

Acknowledgments

All acknowledgements must begin with thesis advisors. Without the help and guidance of Dr. Haizhou Li and Dr. Min-Yen Kan, this thesis could never have been written. Throughout my five years of Ph.D. study, they both not only set a very high research standard, but also showed me vividly what a good researcher should be and do. For that, I am forever grateful. I also owe a similar debt to Dr. Min Zhang of Institute for Infocomm Research who first welcomed me to the field of Statistical Machine Translation. His unwavering support in my early hours of research is invaluable. I am also grateful to have two wonderful thesis committees, Dr. Hwee Tou Ng and Dr. Wee Sun Lee, whose critical questions during my defense help me a lot to iron out the future work of this thesis. I would also like to thank Wang Xi, a linguist whose annotation work makes a bulk of this work feasible. The errors are mine, the thanks are theirs.

Ph.D. life is indeed a lonely journey, but I am grateful to my fellow friends in the Computational Linguistics lab and Web Information Retrieval/Natural Language group who make my journey a pleasant one. I particular enjoy discussing research to Long Qiu, Jin Zhao, Jesse Prabawa, Yee Seng Chan, Shanheng Zhao, Muhua Zhu and Hui Zhang. There is always a joy in the lab although we tease each other (too) often. During my five years in Singapore, I am blessed with many friends inside and outside campus. Listing all of them may fill many pages of this dissertation and understate my appreciation. Thus, let me keep the list in my heart. But, I particularly need to mention Edward Wijaya, who enlighten me in many ways.

I am also blessed to have my family in Indonesia supporting me with full moral support from the beginning to the end. Thanks mom, dad and sis. I owe you so much. The final and utmost acknowledgement should go to the Creator, without whom no part of my life makes any sense.

Untuk Tuhan, Bangsa dan Almamater

To God, Land and Alma Mater

Contents

Abstract

Reordering in Statistical Machine Translation:
A Function Word, Syntax-based Approach

Hendra Setiawan

In this thesis, we investigate a specific area within Statistical Machine Translation (SMT): the reordering task – the task of arranging translated words from source to target language order. This task is crucial as the failure to order words correctly leads to a disfluent discourse. This task is also challenging as it may require in-depth knowledge about the source and target language syntaxes, which are often not available to SMT models.

In this thesis, we propose to address the reordering task by using knowledge of function words. In many languages, function words – which include prepositions, determiners, articles, etc – are important in explaining the grammatical relationship among phrases within a sentence. Projecting them and their dependent arguments into another language often results in structural changes in target sentence. Furthermore, function words have desirable empirical properties as they are enumerable and appear frequently in the text, making them highly amenable to statistical modeling.

We demonstrate the utility of this function word idea in a syntax-based model, which we refer to as the function words, syntax-based (FWS) model, following the recent trend of using syntactic formalisms in modeling reordering. In

demonstrating the utility of the function word idea, we touch and address two problems of the existing syntax-based models, namely: the undergeneration and the overgeneration problems. Our experimental results suggest that our syntax-based approach performs well in the reordering task in perfect lexical choice scenarios where no lexical ambiguities present as well as in the full translation task where lexical noisy interferes, confirming the merit of our function words idea. We also show the virtue of our function word idea when adapted into the state-of-the-art Hiero model in large-scale experiments.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The internet has literally shrunk the world. It connects people from different parts of the world almost instantly. Today, people can easily fulfill their information need, publish their own ideas or communicate with others - all by going to the internet. However, even with this encouraging trend, the internet is still largely fragmented. The hard fact is that internet users come from different linguistic background that forbids them from accessing information written in foreign languages, communicating with foreigners speaking unfamiliar languages and disseminating their ideas to people from different linguistic backgrounds. This fact demands the development of automatic translation systems which can significantly decrease the language barrier, thus providing the much needed access to a large amount of information published in one language to significant parts of the internet population speaking some other languages.

In the guise of Machine Translation (MT) research, the efforts to build automatic translation system have begun as early as late 1940s (Weaver, 1955) and are still ongoing until today. MT's long history serves as a silent witness as to how

challenging the task is. We can find substantial evidence to this claim when we examine how professional translators approach the translation process.

When translators perform their duties, they read the text and rewrite it in the target language. Between reading and rewriting, translators try to comprehend the text by relying on their knowledge about the source and the target language syntaxes, the peculiarities and the idiomatic expressions of the two languages, as well as other linguistics knowledge. More often than not, they have to go beyond what is written to fully understand the text. Efforts to accommodate all these relevant knowledge into the automatic translation process are often considered impractical, since these knowledge are difficult to model and their number is just too large to fit the memory of any current, state-of-the-art computer.

Fortunately, recent advances in Statistical Machine Translation (SMT) research have brought in some optimism. Unlike rule-based systems, SMT focuses only on some parts of the knowledge and treats the translation process as a statistical decision problem. Specifically, it puts the dependencies into real numbers that would be automatically learnt from parallel corpora - collections of translation examples prepared by humans. Benefitting from the growing availability of multilingual corpora and computing resources, SMT researchers have been able to develop statistical translation systems that produce translations of increasingly higher quality, which is adequate to help internet users to get a gist of web contents in unfamiliar languages (e.g. http://translate.google.com).

However by human standards, the output of SMT systems still has many shortcomings. In particular, the translation output often appears out of order and ungrammatical with respect to the target language syntax. The task of arranging the translation output to match the target language order is known as the *reordering task*. This task is extremely challenging and perhaps even as difficult as the translation task itself, since it requires the knowledge of the source and the tar-

get language syntaxes as well as the difference between the two - all of which are either little known or completely unknown to most SMT systems. In this thesis, we focus on addressing this reordering task since better addressing this task would significantly improve translation quality.

The main idea of this thesis is to use the knowledge that hinges on *function words*. The motivation behind this idea is simple. In a great many languages, function words – which include articles, prepositions, auxiliaries, etc. – play important roles in explaining the grammatical relationship among phrases within a sentence. We particularly find a strong support from the *Marker Hypothesis* (Green, 1979), which states that natural languages are "marked" for syntactic structure at surface level, implying that there exists a closed set of words or morphemes that appear in a very limited set of grammatical context. In some languages, such set corresponds to function words.

We can also find more support for this function words idea from the concept of syntactic heads in linguistics theory. The syntactic head refers to a lexical entity that determines the syntactic categories of the phrase of which they are the member. Although it is a matter of debate, there is a recent tendency toward equating function words as heads of phrases. For instance, Abney (1987) suggested the use of determiner as the head of a noun phrase in his Determiner Phrase analysis, as opposed to the traditional way of equating noun as the head. In a number of languages other than English, function words are also known to play pivotal roles in the syntax. For instance, in Japanese and Korean, function words appear in most, if not all, phrases, acting as case markers.

When we casually inspect data, we often see that projecting function words and their arguments often results in a structural change in the realized sentences. As a reference, Chinese function words involve in almost all the hand-crafted transformational rules used to reorder the input Chinese sentence into the English order

as defined in (Wang, Collins, and Koehn, 2007).

Moreover, function words also have many desirable empirical properties. First of all, the member of this class of words is enumerable as it rarely accepts new members. Furthermore, the frequency of function words in the corpus is also very high, which eventually makes them easy to identify and more amenable to statistical modeling.

In implementing this function word idea, we follow the recent syntax-based approach. Specifically, we focus on a class of syntax-based approaches, namely: *formally syntax-based* (FSB) approach. The FSB approach is unique, since it uses a syntactic formalism that is not necessarily guided by any particular linguistic theory, thus requires no linguistic annotation. We decide to focus on this approach not only because it is simple and some of the state-of-the-art SMT systems, in fact, belong to this class of approach, but also because we believe that the full benefit of the function words idea can be better demonstrated in such a knowledge poor environment. Nonetheless, the idea presented in this thesis may also be applicable to other strand of SMT approaches, although it is not explored in this thesis.

One can think of our approach as a foreign language learner who has a limited knowledge about the target language grammar but he or she is quite knowledgeable about the role of function words. Such a person should be able to make an educated guess about the target language order by looking at the function words alone. Throughout this thesis, we refer to this proposal as the **function word, syntax-based (F W S) approach**. In summary, the F W S approach is developed into a specific variant of SCFG, which we call the head-driven SCFG where the heads are equated with function words, and several statistical models inspired by the function word idea. Note that since we decide to focus on a knowledge-poor environment, the definition of function words may not always conform to any linguistic sense.

In this thesis, we also demonstrate the contribution of our function word idea

in better addressing two important problems of FSB models: the overgeneration and the undergeneration problems. In the coming Section 1.2, we discuss how the design of the existing FSB models results in the overgeneration and the undergeneration problems. In Section 1.3, we discuss the F W S model and describe how in principle this model can address the two aforementioned problems. In Section 1.4, we end this chapter with the guide to this thesis.

1.2 Overgeneration and Undergeneration

The recent move to syntax-based models has enabled SMT models to efficiently address difficult reordering problems, such as certain non-local reorderings that are deemed computationally too challenging for their predecessor, phrase-based models (Koehn, Och, and Marcu, 2003). Unlike phrase-based models, syntax-based models view the translation process as a joint process of generating a sentence pair from smaller phrase pairs via the application of recursive, bilingual rewrite rules; creating an intermediate hierarchical structure that resembles natural language syntax. Modeling long-distance reordering is simple for syntax-based models, since they treat long and short distance reorderings identically as rewrite rules, thus modeling different kinds of reordering requires no additional parameter.

Depending on the source of knowledge from which rewrite rules are learnt, syntax-based models can be broadly categorized into two classes: *formally syntax-based* (FSB) and *linguistically syntax-based* (LSB) models. The latter learns rewrite rules from parallel text with some linguistic annotation, thus the learnt rules are fully adherent to some linguistic theories; while the former learns rewrite rules from plain parallel text without any annotation, thus the learnt rules are not necessarily in any linguistic sense. In this thesis, we adopt the FSB approach as it represents the most realistic scenario since the majority of parallel corpora comes without any annotation and we believe that the benefit of function words idea can be better

demonstrated in this knowledge-poor approach.

In committing to the knowledge-poor approach, our main goal is to advance the FSB models without the help of linguistic annotation. To achieve this goal, we first identify problems that are common to the existing FSB models and focus our effort to better address these problems using the function word idea.

Formally, all FSB models come in guise of *Synchronous Context Free Grammar* (SCFG) (Aho and Ullman, 1969), which is a generalization of Context Free Grammar to bilingual cases. In their abstract level, SCFG rules takes the following generic form:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle \tag{1.1}$$

where X is a nonterminal symbol while γ and α are the strings in the source and target languages, respectively. The $\sim$ symbol indicates the correspondences between symbols in γ and α, typically expressed via co-indexation.

Translating a source sentence for an FSB model is equal to applying a set of rules in a certain order of application to cover all words in the source sentence, producing a hierarchical structure, which is often known as *derivation*. The translation of a source sentence is then obtained by simply reading-off the target side of the derivation.

Recently, there has been a growing interest to improve FSB models by introducing syntactic information to the model, effectively relaxing the knowledge approach. For instance, Zollman and Venugopal (Zollmann and Venugopal, 2006) tries to introduce syntactic constraint from target language syntax into the original FSB model. In this thesis, we hypothesize that we can improve the FSB model while still maintaining the knowledge poor assumption via our function word idea.

Ideally, given a particular source sentence, an FSB model should suggest only one derivation, i.e. the one gives the correct target reordering. However in practice, the model often fail to generate the correct derivation or even it does, it sometimes

generates several incorrect ones. We use the terms *overgeneration* and *undergeneration* to refer to these problems, as they are well known especially in monolingual parsing community. Hence in reordering sense, the overgeneration problem refers to cases where the model generates more derivations than appropriate for a given source sentence; meanwhile, the undergeneration problem refers to cases where the model fails to generate the one derivation that gives to the correct reordering.

The overgeneration and the undergeneration problems can be attributed to many factors, including those related to the genuine ambiguity of the languages. This means that eliminating these two problems is not a reasonable aim. However, there are some other causes that are due to the characteristics of the model, which we intend to focus on, especially those that are related to the design of the Hiero model – the state-of-the-art FSB model (Chiang, 2005). Before discussing which characteristics are problematic, we first briefly review the characteristics of the Hiero model below.

Rules in the Hiero model follow the generic form described in Rule 1.1 with several unique characteristics. First of all, Hiero rules comes only with one type of nonterminal symbol, hereafter, referred to via the X symbol. Secondly, the source and target language strings (γ and α respectively) in Hiero rules consists of a combination between nonterminals (Xs) and lexical items (individual word and even multi-word). This characteristic allows Hiero to capitalize on the phrase-based approach's strength of modeling multi-word translation. Lastly, the correspondences ($\sim$) between the source string (γ) and the target string (α) are established only on one-to-one basis and only between nonterminals.

Mainly for efficiency reason, Hiero also imposes several constraints, such as limiting the maximum number of nonterminal to two and forbidding the creation of rules with adjacent nonterminals. We are specifically interested in the second constraint, which we will subsequently refer to as the *non-adjacent nonterminal*

constraint. Rule 1.2 below is one example of a valid Hiero rule.

$$X \rightarrow \langle \text{ 电脑 和 } X, \text{ computers and } X \rangle \tag{1.2}$$

Which of the above characteristics may cause the overgeneration and the undergeneration problems? We focus on three characteristics and discuss them in more detail below. As throughout this thesis we consider the Hiero model as the representative of the FSB models, we will consider the above characteristics as the characteristics of the FSB models in general.

- **The use of only one type of nonterminal symbol (X).** In theory, rewrite rules can have as many types of nonterminal symbols as possible and ideally, these types should correspond to some linguistic categories. However, due to the lack of exposure to linguistic annotation among many other reasons, rewrite rules in FSB models come only with one type of nonterminal symbol. Such a homogenous use of the generic nonterminal symbol X, unfortunately, is the main source of the overgeneration problem since it gives a maximum flexibility that allows the model to generate many different derivations from the same set of rewrite rules; many of which unfortunately would lead to incorrect translations. Overgeneration can be curb either by imposing constraints, lexical items or developing strong models to reliably select the correct derivation. In terms of the latter, the homogenous use of X leaves the model only with the standard treatment via intersecting the grammar with n-gram language model. This is suboptimal because it only looks at the target language side and local information.

- **The fine-grained modeling of lexical items.** To curb the overgeneration problem incurred by the homogenous use of the single nonterminal symbol, FSB models introduce lexical items to their rewrite rules. This effectively reduces the number of possible derivation for a given source sentence. While

beneficial, the lexical items are introduced into rules in an agnostic manner, ignoring the fact that lexical items may come from different lexical categories. As such, both content words as well as function words are modeled identically in a fine-grained manner. Unfortunately, in modeling content words, FSB models may run into data sparsity issues since unlike function words, these words appear in low frequency in training data. In some cases, modeling content words might even be detrimental, because these words tend to have different syntactic behavior depending on their context. The incurred low generalization power would ultimately lead to the undergeneration problem, since a slight lexical mismatch can make all rules learnt from training data inapplicable to unseen test sentences, providing the model with inadequate set of rules to generate the correct derivation.

- **The use of non-adjacent nonterminals constraint.** In addition to the overgeneration and the undergeneration problems, FSB models have to deal with spurious ambiguity, which refers to a situation where the model generates many derivations that lead to the same translation (Chiang, 2005) – regardless of whether the translation is the correct or the incorrect one. This ambiguity is often perceived as a decoding problem, as it introduces an undesirable crowding effect that complicates the decoding process (Liang and Klein, 2008). To curb this ambiguity, Hiero forbids the creation of rules which are deemed to be the major source of the ambiguity, i.e. rules with adjacent nonterminals, by employing the *non-adjacent nonterminals* constraint. Unfortunately, this constraint reduces Hiero's generalization power, as posited by Menezes and Quirk (2007) since it limits the model's ability to generalize content words. This eventually aggravates the undergeneration problem, as this constraint may filter outs rules that are essential to correctly translate some unseen test sentences.

In principle, the overgeneration problem (i.e. caused by the homogenous use of one type of nonterminal symbol) is attributed to the fact that most of the work in FSB models are inspired by *Inversion Transduction Grammar* (ITG) (Wu, 1997). Although for ITG, overgeneration is an essential feature rather than a problem, as its main purpose is for bilingual analysis, i.e. to verify the validity of a particular reordering. Meanwhile, the undergeneration problem can be seen as undesirable negative effects from the FSB models' efforts to combat the overgeneration problem since these efforts (both the fine-grained modeling of lexical items and the use of non-adjacent nonterminals constraint) limits the model's ability to learn essential rules useful for creating the correct derivations for some unseen sentences.

1.3 Function Word, Syntax-based Approach

Here, we argue that our function words idea has largely-unexplored potentials that can be used to better address the overgeneration and the undergeneration problems of the existing FSB models without relying on linguistic knowledge. We develop this idea on top of a formalism which we call the head-driven Synchronous Context Free Grammar (head-driven SCFG), extending SCFG to include the notion of head. The detail definition of this grammar will be discussed in Chapter 3 but a high level overview is discussed here.

In a nutshell, the head-driven SCFG differs from the existing models in several respects:

1. The head-driven SCFG comes with *two types of nonterminal*: Y and X, where the former is used to denote the **heads** while the latter to denote the **arguments**. An argument is basically any span of text whose reordering is influenced by a head, where the head is equated with function words in our implementation to reflect the main idea of this thesis. In essence, this

grammar is inspired by a linguistic insight that words in a phrase are organized around its head (Radford, 1998).

2. The head-driven SCFG views the expansion of rules as *a head-outward process*, following Collins parsing model (Collins, 2003) where the head is considered to be generated first and arguments are then generated one by one starting from the one closest to the head.

3. The head-driven SCFG *lexicalizes* nonterminals with the information about the heads (hereafter head-lexicalization), propagating such information from lower level of the hierarchical structure to its higher level. Thus, in our syntax-based model, the nonterminals carry a richer set of information than its counterpart in the existing models.

How can a head-driven SCFG, in which heads are equated with function words, better addresses the overgeneration and the undergeneration problems of the existing FSB models? First of all, a head-driven SCFG can potentially address the overgeneration problem caused by the homogenous use of the generic nonterminal symbol since the model now contains two types of nonterminals and lexicalizes the nonterminals that can be used to develop statistical models to select the correct derivation. Second of all, a head-driven SCFG can also address the undergeneration problem caused by the fine-grained modeling of lexical items since it focuses on modeling function words that theoretically corresponds to words with high generalization power. Finally, a head-driven SCFG can also address the undergeneration problem due to the non-adjacent nonterminals heuristic since it effectively relaxes the constraint by modeling the expansion of a rule as a head-outward process.

We develop the F W S approach in two stages, resulting in the basic Function Word, Syntax-based (F W S) model, which we have reported in (Setiawan, Kan, and Li, 2007) and the improved F W S model. In the basic F W S model, we

concentrate on the feasibility of the F W S approach and focus on developing the F W S idea into several *stateless* statistical models, which looks at no contextual information. Meanwhile in the improved model, we focus on developing the F W S models into *stateful* statistical models, which looks at rich contextual information[1].

1.4 Guide to the Thesis

The remainder of this thesis is organized as follows:

Chapter 2 reviews the related work on SMT starting from early models to the more recent ones, focusing on their reordering components. In this chapter, we review the issues that the current state-of-the-art models have and have not addressed, expanding the discussion in Section 1.2.

Chapter 3 provides a general overview of the proposed function word, syntax-based reordering. In this chapter, we develop the detail formalism of the head-driven SCFG. More importantly, this chapter serves as a preview for understanding the main part of this thesis in Chapters 5 through 8.

Chapter 4 describes the setup for the experiments conducted in this thesis along with the detail of the baseline systems. In this chapter, we also describe a pilot study to investigate about whether we can rely only on the knowledge embedded in function words to reorder sentences.

Starting from Chapters 5 through 8, we present the Function Word, Syntax-based (F W S) model, implementing the components discussed in Chapter 3. In Chapter 5, we discuss the basic F W S model - a natural entry point to the overall framework. Here, we focus on assessing the feasibility of the F W S approach. In this chapter, we provide error analysis of the basic F W S model, which motivates

[1]An example of stateless model is the translation probability in the standard phrase-based model, while an example of the stateful model is the n-gram language model, which estimation requires the previous $n-1$ words.

the development of the subsequent models.

In Chapter 6, we focus on developing a variety of techniques to identify function words. In Chapter 7, we propose an argument selection model as a way to address the undergeneration problem, which is due to the non-adjacent nonterminal constraint. Meanwhile, in Chapter 8, we focus on addressing the overgeneration problem by proposing a pairwise dominance model utilizing the lexicalization provided by the head-driven SCFG. Chapter 9 describes the complete experimental results and discusses error analyses of the improved F W S model, which is the combination of the proposals developed in Chapters 6 through 8. We also show that the virtue of the function word-based reordering idea extends by adapting some statistical models into the state-of-the-art Hiero model in Chapters 10 and show that the Hiero model can benefit from the adapted models in a large-scale experiments. We end this thesis in Chapter 11 where we summarize its work, recapitulate its contributions, point out its limitations and lay out future directions.

Chapter 2

Related Work

Given a translated sentence still ordered in the source language order, the ultimate goal of a reordering model is to assign a new location to the translation of each word so that the reordered translation matches the target language order. This chapter reviews the previous and the current state-of-the-art SMT models particularly in terms of the reordering model they employ. Specifically, we look at some key issues that have been and have not been tackled by the existing reordering models.

In our review, we discuss the existing models in chronological order, starting from the first generation word-based models, to the phrase-based models and to the more recent syntax-based models, expanding the discussion in Section 1.2. Readers who are already familiar with SMT models may want to go directly to Section 2.3.2, where we discus the key issues addressed by this thesis.

Throughout this chapter, we use the Chinese to English translation illustrated in Fig. 2.1 as our running example. For convenience, we consistently use the terminologies of the distributional hypothesis (Harris, 1954) – although the actual models may not use the same terminology or form – which views a reordering model as a model that estimates the following formula $P(pattern|unit, context)$ where *unit* represents the linguistic entity being moved, *pattern* refers to the param-

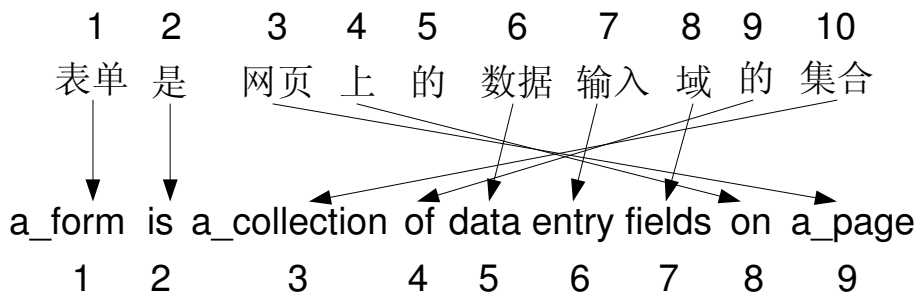

Figure 2.1: An illustration of how words move when translated.

eters over the unit's new location and *context* defines the circumstances in which the unit moves to the new location specified by the pattern. The definition and estimation of these three components, as shown throughout this chapter, dictate the performance of the models.

2.1 Word-based Approach

The first generation word-based models, of which the IBM model series (Brown et al., 1993) is the pioneer, define the granularity of the unit at the individual word level. These models rely on positional information in modeling word reordering. More specifically, they tie the unit's parameter to the position of the word being moved in the source sentence and the pattern's parameter to the word's new location in the target sentence. For instance, the movement of the word 网页 (a page) in Fig. 2.1 is formulated as $P(j{=}9|i{=}3)$ where i is the word's original position on the source side while j is the word's new location. Although simple, this formulation is unfortunately suboptimal in several respects.

First of all, such reordering models are insensitive to the identity of the unit and, let alone, the context in which the unit moves. The most sophisticated IBM model (model 5), to a certain extent, addresses the first issue by conditioning

the pattern on the word's automatically-obtained class, while the HMM alignment model (Vogel, Ney, and Tillmann, 1996) partially addresses the the second issue by conditioning the pattern on the previous word's new location. Toutanova et al. (2004) combined these two pieces of information together and showed that the combination improves the word alignment quality.

Second of all, tying the parameters to the positional information may not generalize well since the position of the same word tends to be different across different sentences. One can easily come up with many other sentences where the word 网页 (a page) appears not at the third position in the sentence. Furthermore, such a parametrization also complicates the modeling of the long-distance reordering phenomenon since the models would have to introduce (i,j) pairs which size grows exponentially with respect to the distance the unit may travel (Och and Ney, 2003). Knight (1999) showed that allowing words to move freely to any position is equal to solving an *NP*-hard problem, intractable even for current state-of-the-art computers. To curb such a high computational complexity, the word-based models often limit the maximum distance a word may travel (Berger, 1996) and rely on approximations such as (Germann, 2003; Och, Ueffing, and Ney, 2001; Germann et al., 2001), thus incurring the corresponding loss in modeling long-distance reordering.

2.2 Phrase-based Approach

Learning from the weaknesses of the word-based approach, the phrase-based approach improves statistical machine translation formulation in at least two respects.

First of all, the phrase-based approach extends the granularity of the unit to account for spans longer than one word, grouping several word translations into one cohesive translation unit, which hereafter will be referred to as *a phrase translation* (also known as a bilingual phrase). This phrase unit may not be a phrase in any linguistic sense since the extraction process relies from parallel corpora without any

genuine segmentation information using the *consistent alignment* heuristic (Och and Ney, 2003) below.

$$\mathcal{PT}(f_1^J, e_1^I, A) = (f_j^{j+jj}, e_i^{i+ii}) : \forall (i', j') \in A : j \leq j' \leq j + jj \leftrightarrow i \leq i' \leq i + ii \tag{2.1}$$

where $\mathcal{PT}$ stands for phrase translations, f_1^J and e_1^I are the source and target sentences of length J and I respectively, A is a set of alignments (i', j') between f_1^J and e_1^I and i and j are used to indicate source and target word indexes respectively. The consistent alignment heuristic basically specifies that a source phrase (f_j^{j+jj}) of length jj and its translation e_i^{i+ii} of length ii is a valid phrase translation if the source phrase is only aligned with the words inside its translation. Note that we will reuse this consistent alignment heuristic in the parameter estimation of our models.

Fig. 2.2 shows an example of how a phrase-based model would translate the example in Fig. 2.1. Even without such information, the phrase-based models benefit greatly from the introduction of this phrase translation, since it enables the models to remember short-distance reordering phenomena that appear in the training data. Here, the phrase-based model effortlessly captures the swap between the word 网页 (a page) and the word 上 (on) since it has been memorized in a phrase translation unit – the third one. In many evaluation exercises, relying on such phrase translation unit has enabled the phrase-based models to outperform the word-based models, as demonstrated by the Pharaoh system (Koehn, 2004a).

Secondly, the phrase-based approach simplifies the parametrization of the pattern from the position-based parametrization to the orientation-based one. Tillman (2004) introduced a three-valued orientation values: Left, Right and Neutral. The Left value refers to the reordering pattern where the current phrase translation

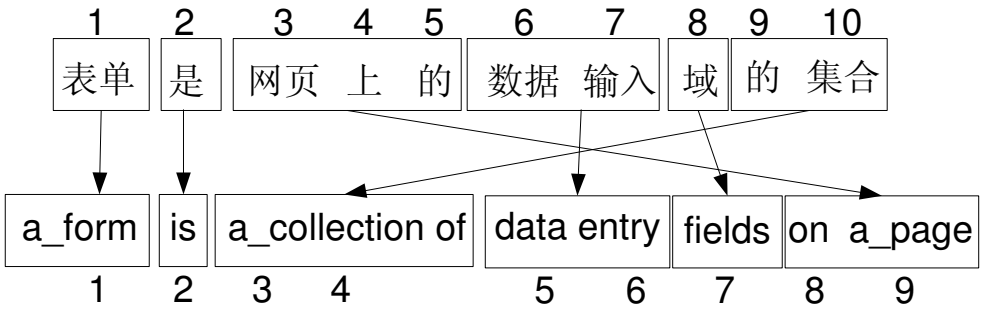

Figure 2.2: An illustration of how phrases move when translated.

under consideration ends up on the left, before the preceding one[1], while the Right refers to the other case where the current phrase translation ends up on the right, after the preceding one. The Neutral value refers to a special case where there is another phrase translation in between the current and the preceding phrase units. According to this parametrization, the orientation value for the phrase 域 (fields) is Right, because its translation appears after the translation of its preceding phrase 数据 输入 (data entry).

Partly because of this simpler set of parametrization, the recent reordering models can now afford a richer parametrization for the unit as well as for the context. For instance, Tillman and Zhang (2005) introduced the Unigram Block model while Kumar and Byrne (2005) introduced the Local Phrase Reordering model; both of which basically use the lexical identity of the unit in the model. This simple idea has been adopted by the current state-of-the art phrase-based Moses (Koehn et al., 2007) system and has shown to significantly outperform its predecessor, the Pharaoh system. In this unigram model, the movement of the phrase 域 (fields) is in the form of P(orientation=Right|unit=域 (fields)). Note

[1]For consistency with subsequent discussions, we deliberately define the notion of the preceding phrase along the source side. Readers should note that its actual definition may be defined along the target side depending on the decoder implementation. Nevertheless, the idea is clear regardless of the actual definition of this phrase since the orientation value is symmetric.

that now, there are a separate statistics for each phrase.

Also partly because of this simpler set of parametrization, more recent models are now able to afford a more complex context modeling. For instance, Tillman and Zhang (2005; 2007) introduced the Bigram Block model which considers the lexical identity of the preceding phrase translation as context. Along this same idea, there are also some other proposals, such as (Zens and Ney, 2006; Nagata et al., 2006; Al-Onaizan and Papineni, 2006) that differ from each other with regard to the estimation of the context. Unfortunately, although these efforts enable phrase-based models to address the word-based approach's concerns, these models are still problematic in several respects.

First of all, the long-distance reordering is still difficult to accommodate. In particular, the models use the orientation-based parameters, which even though simpler, still rely on positional information as a result, these models do not generalize well.

Secondly, the flexible definition of the phrase translation unit creates lots of modeling problems. For instance, such flexibility can make the orientation value of a phrase unit to be different across context. For instance, the orientation value of the phrase 的 集合 (a collection of) at the end of the source sentence is Left if the preceding phrase unit is a three-words phrase 数据 输入 域 (data entry fields) but Neutral if the preceding phrase unit is a one-word phrase 域 (fields).

Thirdly, the rigid definition of context, i.e. always the preceding phrase, is suboptimal. For instance, the context for the phrase 的 集合 (a collection of) at the end of the source sentence linguistically should be the whole head noun phrase 数据 输入 域 (data entry fields), which spans two phrase translation units in Fig. 2.2. Meanwhile, the context for the phrase 网页 上 (on a page) naturally is the succeeding phrase rather than the preceding one.

Lastly, the models are heavily lexicalized, thus susceptible to the sparse data

issue. For instance, modeling the swap between the word 网页 (a page) and the word 上 (on) is not useful to model other cases of post-positional to pre-positional shift. Likewise, memorizing the lexical identity of the context may also not be useful since the context of the same unit tends to have different wording in different sentences.

SMT researchers have long acknowledged these problems. Ideally, the phrase movement should be driven by syntactic principles rather than lexical level information. The Moses system has provided a framework, known as the factored translation model (Koehn and Hoang, 2007), that allows the translation process to exploit richer set of linguistic information (e.g. lemma and morphological features). However, incorporating syntactic information into the phrase-based framework remains an open problem.

To date, current efforts to incorporate syntactic information to phrase-based models have met limited success – some even lead to performance deterioration. For instance, Koehn et al. (2003) reported that restricting the phrase translation unit only to that of syntactic phrase harms the performance. Birch et al. (2007) experimented with rich syntactic information, such as part-of-speech (POS) tags and supertags taken from Combinatorial Categorical Grammar (CCG) lexicons, however, their experiments showed that using such linguistically-rich information leads to no significant improvement when compared to the unigram lexicalized reordering model.

2.3 Syntax-based Approach

The move to syntax-based approach allows some of the phrase-based models' concerns to be addressed elegantly. This approach views the reordering process as the application of a series of bilingual rewrite rules, which recursively builds a hierarchical structure that resembles natural language syntax. In terms of the definition of the unit, the syntax-based approach uses a special phrase translation unit, to

which we subsequently refer to as a *nonterminal translation* (or nonterminal in short). Different from the phrase translation, the nonterminal is *typed* (associated with a label, thus equals to other phrases sharing the same label) and *nested* (may be formed in a several intermediate steps, indicated by a subtree covering a phrase covering a certain span of text).

In using rewrite rules, the syntax-based approach makes a *domain of locality* assumption that specifies the contextual dependencies (and the independencies) of a phrase. Two phrase translations are considered dependent if they share a common parent; but independent if they do not share a common parent. This domain of locality is particularly desirable for the pattern parametrization as well as the context modeling. In terms of the former, the pattern parameters are defined locally within the confine of a node, thus are no longer tied to positional information. In terms of the latter, the definition of context is no longer tied rigidly to the preceding phrase, but rather flexibly depending on the position of the sibling nodes.

However, developing a syntax-based model is non-trivial since it depends on the models' ability to induce the grammar rules in a situation that is far from ideal. In an ideal situation, syntax-based models expect parallel corpora that are aligned at phrasal level as the input to the grammar induction process, where both sides of the corpora come with hierarchical structure and are connected to each other via nodes in their respective structure. From such an ideal input, the grammar induction process can then just read off the grammar rules in a relatively straightforward manner. Up to now there has been no such constituent-aligned corpora available in a significant amount to the community, and manually constructing one would be a daunting task since it involves dealing with the complexities of a pair of languages (for a survey of ongoing work, see (Rambow et al., 2006)).

Without constituent-aligned corpora, researchers must accept more realistic scenarios: relying on unannotated parallel corpora alone or on parallel corpora with

some partial linguistic annotations. This roughly divides syntax-based models into two groups (Chiang, 2005)[2]: *formally syntax-based* and *linguistically syntax-based* models. The former takes the *knowledge poor* path, inducing the grammar entirely from unannotated parallel corpora; while the latter takes the more *knowledge rich* one, inducing the grammar from parallel corpora that first need to be annotated with the parse trees at either or both sides of the parallel text.

Subsequently, since we implement our idea in the knowledge-poor path, we will focus more on the formally syntax-based approach. But for completeness sake, we first briefly cover the linguistically syntax-based approach to show several key issues that differentiate this strand of approach from the other one.

2.3.1 Linguistically Syntax-based Approach

The linguistically syntax-based (LSB) approach assumes that the parallel text is annotated with some linguistic information either on the source language, on the target language or on both languages. The models that subscribe to this approach attempt to capture linguistically-motivated learning bias embedded in the annotation, using a syntactic formalism that is guided by a syntactic theory.

The first syntax-based model (Yamada and Knight, 2001) views the translation process as a tree transformation process. It expects the input sentence to be annotated with the syntactic parse tree and reorders the text via reordering and insertion operations over the parse tree. Although it performs better than word-based systems, its performance is surprisingly lower than the simpler phrase-based systems. This result runs counter with the intuition about the potential benefit of having linguistically-motivated information, hinting that incorporating such a deep

[2]There has been no consensus about these terminologies as of this thesis writing. For instance, the formally syntax-based model is also known as syntax-inspired, while the linguistically syntax-based model is also known as syntax-directed. However, the distinction between these two approaches is consistent.

syntactic information into the reordering process is not trivial.

The benefit of annotating parallel text remains an open question for a few reasons. First of all, the syntactic parse tree is typically obtained through an automatic process, thus not perfect. Secondly, the estimation of the syntactic parse tree is independent from the estimation of other components, thus mismatches are very likely to occur. Thirdly, the genuine structural difference between the two languages often makes it impossible for the model to get to the correct target language order by using only simple node reordering operations.

The complexity of integrating deep syntactic knowledge has also been extensively studied. For instance, Fox (2002) showed that complex bilingual rewrite rules are necessary even for a language pair that comes from the same language family. Wellington et al. (2006) showed that syntactic parse tree imposes additional linguistic constraints that greatly reduce the ability of syntax-based models to induce rewrite rules from the training examples. In the light of these issues, some researchers have proposed several solutions along several different lines.

The most popular approach to address these issues is by employing a more expressive grammar. One of the most widely-used formalism is the tree-transducer formalism[3] where rewrite rules store information about a parent node together with all its successor nodes down several levels to the leaf nodes. Depending on which side contains the syntactic information, these models employ different variants of tree transducers, such as: 1) tree-to-string models (Liu, Liu, and Lin, 2006; Quirk, Menezes, and Cherry, 2005), which assume a parse tree on the source side; 2) string-to-tree models (Galley et al., 2004; Marcu et al., 2006), which assume a parse tree on the target side; and 3) tree-to-tree models (Cowan, Kučerová, and Collins, 2006; Zhang et al., 2007; Zhang et al., 2008), which assume the parse trees on both sides. Some other solutions have also been proposed in the guise of the so-called tree-

[3]For a survey, see (Knight and Graehl, 2005).

sequence model (initially called the forest model) (Liu et al., 2007) that includes allowing rewrite rules to model a sequence of nonterminals.

In parallel, some researchers have also proposed to tackle the problem from a different point of view: addressing the tension between the word alignment and the syntactic parse tree that is caused by the fact that the two are generated independently from two noisy processes. For instance, Cherry and Lin (2006) and DeNero and Klein (2007) attempted to reconcile the tension by integrating syntactic information into the alignment process.

Models in this strand of approach also suffer from the overgeneration and the undergeneration problems. However in linguistically syntax-based models, these problems are mainly due to the genuine ambiguity in languages, rather than due to the design of the grammar and they partly have been taken care of by the use of linguistically-motivated phrase categories.

2.3.2 Formally Syntax-based Approach

The formally syntax-based (FSB) approach arguably represents the most realistic strand of syntax-based approach. This strand of approach assumes minimal information possible, relying only on the parallel corpora without any linguistic annotation to extract rewrite rules. Without any linguistic information, however, such syntax-based models face a more challenging task since they work with a larger set of unknown information than their counterpart linguistically syntax-based models.

To estimate unknown information, FSB models make several assumptions, especially to approximate the shape and the content of the hierarchical structures between the source and target sentences. All the FSB models that we review here come in the guise of the Synchronous Context Free Grammar (SCFG) formalism, previously known as the syntax-directed translation system (Aho and Ullman, 1969), which is the generalization of the Context Free Grammar (CFG) to the bilin-

gual case. Different from the tree transducer formalism, the rewrite rules in SCFGs only store the information about the parent node (nonterminals on the rules' left hand side) and its immediate children (nonterminals on the rules' right hand side), forcing the source and target parse trees to be *isomorphic*, i.e. aligned at every node.

Additionally, FSB models typically follow the Inversion Transduction Grammar (ITG) hypothesis (Wu, 1997), which assumes that the possible hierarchical structures (also known as derivations) between the source and target sentences are those that are *binarizable*, i.e. transformable to another hierarchical structure where all the parent nodes have exactly two children nodes. This assumption directly defines the shape of the possible hierarchical structures and confers syntax-based models a desirable computational property. Some studies (Zens and Ney, 2003; Wu, 1997; Wu, Carpuat, and Shen, 2006) also show that this assumption is indeed reasonable for many language pairs, such as Chinese-English and Arabic-English.

Theoretically, syntax-based models can come with as many nonterminal labels as possible. Ideally, these labels should correspond to some linguistic sense. However, without access to linguistically-motivated information, FSB models can only afford to use one generic type of nonterminal that is typically labeled as X. Note that unlike the syntactic category used in linguistically syntax-based models, this symbol imposes no constraint on what kind of text span can be denoted as X.

Unfortunately, the decision to use only this generic symbol causes the FSB models to overgenerate, producing more derivations than appropriate. In particular, the homogenous use of the generic nonterminal suggests that the parent nonterminal on the LHS and its children on the RHS are identical, imposing no constraint about the correct order of application for the rule. As such, one rule can be applied in a flexible manner with an equal probability, i.e. before or after another rule.

A standard solution to address the overgeneration problem often involves

intersecting the grammar with the target n-gram language model (Zollmann and Venugopal, 2006). Thus, the correct order of rule application corresponds to the most probable surface translation. However, this partial solution is suboptimal since it only looks at local information and on the target side only. On top of employing language model, most successful proposals to curb the overgeneration problem involve the introduction of lexical items into rewrite rules, using information from noth the source and target languages.

Lexicalized ITG (LITG) and BiLexicalized ITG (BLITG) models use lexical items through what we call the *lexical propagation* method. These grammar first assume that there is one special token called the head in a sentence and then propagate the information of this head from lower level structure to higher level structure, equating the head as the backbone of the hierarchical structure. In both models, the parent node contains two types of children: the head node and the modifier node where the former is propagated from the lower level structure to the parent node through the head node but the latter is not. BLITG differs from the LITG with respect to the modifier node where the former associates the modifier node with a lexical item while the latter does not.

Rule 2.2 represents an example of LITG rule while Rule 2.3 represents an example of BLITG rule.

$$X(h) \rightarrow \langle X_1(h)X_2, X_2X_1(h)\rangle \tag{2.2}$$

$$X(h) \rightarrow \langle X_1(h)X_2(m), X_2(m)X_1(h)\rangle \tag{2.3}$$

where h refers to the head word, and m refers to the lexical item heading the modifier node that is not propagated. Note that the nonterminals are co-indexed to indicate reordering and not to introduce new type of nonterminals.

These two grammars offer a promising idea since their method of introducing lexical items provides an elegant way to address the overgeneration problem. Specifically, it offers a rich set of information that can potentially be used to select

the correct derivation. Our proposed head-driven SCFG, to some extent, draws its inspiration from these two grammars. However, these two grammars are currently designed as alignment models, thus, they cannot be directly used to address the reordering task – at least not until they resolve the remaining non-deterministic factors, such as which word should become the head word and which children node represents the head node.

Meanwhile, the Bruin model (Xiong, Liu, and Lin, 2006) uses lexical item in a method which we call *nonterminal features*. In particular, this model is essentially a Maximum Entropy (ME) model (Berger, Pietra, and Pietra, 1996) where lexical items are used as ME features to make a decision which reordering rule to be applied at a certain context. More concretely, the Bruin model consists of the following two rules, which are the rules of the Bracketing Transduction Grammar (BTG) (Wu, 1997):

$$X \rightarrow \langle X_1 X_2, X_1 X_2 \rangle \tag{2.4}$$

$$X \rightarrow \langle X_1 X_2, X_2 X_1 \rangle \tag{2.5}$$

and to decide whether Rules. 2.4 or 2.5 should be to applied, Bruin takes the lexical items at the borders of X_1 and X_2 as the main features.

Finally, the state-of-the-art Hiero system uses lexical items through what we call *RHS lexicalization*. More concretely, Hiero introduces the nonterminals, which are known as the hierarchical phrases. In these hierarchical phrases, Hiero introduces lexical items into the rule's RHS. Rule 2.6 below represents one example of hierarchical rule that can be extracted from the example in Fig. 2.1.

$$X(a) \rightarrow \langle X_1 \text{ 数据输入域 } X_2, X_2 \text{ data entry fields } X_1 \rangle \tag{2.6}$$

The ability to combine generic nonterminal symbols and lexical items is often considered as the Hiero system's main strength since it enables Hiero to accommodate

non-contiguous phrases and to emulate the phrase-based approach's strength of remembering short-distance reordering phenomena.

Empirical results show that these proposals are able to address the overgeneration problem. However, we argue that there are still rooms for improvements since in addressing the overgeneration problem, since 1) the method still contains unresolved non-deterministic factors (in case of LITG and BLITG), 2) the method only uses local information (in case of Bruin); and the method causes the model to undergenerate (in case of Hiero).

More importantly, common to these proposals is that they introduce lexical items agnostically, ignoring the fact that most of these lexical items belong to content word class that is not particularly amendable to statistical modeling. Content words appear rarely in the corpus and often have different behavior in different context. Modeling content words unfortunately may create the sparse data concern, as such it can prevent the models to generate the derivation that leads to the correct reordering.

Specific to the Hiero model, undergeneration is aggravated by the *non-adjacent nonterminals* heuristic. Essentially, this heuristic is employed by the Hiero model to forbid the creation of rules with adjacent nonterminals, which are deemed as the main source of *spurious ambiguity* (Chiang, 2007). This ambiguity refers to cases where many derivations with the same probability lead to the same surface translation and it is highly undesirable for its crowding out effect (Liang et al., 2006) especially in the approximate decoding setting. However, as posited by Menezes and Quirk (2007), this heuristic again reduces the generalization power of the system, since it limits the model's ability to generalize content words only in certain patterns.

2.4 Summary

The goal of this chapter is twofold. First, it provides an overview of the existing work that address the reordering task. Second, it provides a background information which relates our proposed F W S approach with other existing work. In our review, we discussed the existing models in the chronological order, starting from the first generation word-based models, to phrase-based models and to the recent syntax-based models. In particular, we discussed the issues of the earlier models and showed how the more recent models address them.

Our proposal is most closely related to the formally syntax-based models discussed in Section 2.3.2, which assume minimum information possible in learning the rewrite rules. We showed that the main characteristic shared by the existing formally syntax-based models is that all the nonterminals are labelled uniformly with a single label X. We emphasized that this assumption is problematic because it makes the model overgenerates. We also showed that the current efforts to address this issue are still suboptimal since they mostly rely on lexical level features, which has generalization concerns and often makes the models undergenerate. As mentioned earlier, we hypothesize that our proposal, which we will develop shortly, can better address these two problems.

Chapter 3

Function Word, Syntax-based Reordering

Here, we describe our function word, syntax-based (F W S) approach, specifically its formalism: the head-driven SCFG. In Section 3.1, we start with a recap about the three differences between the head-driven SCFG and existing SCFGs, which will then lead to a discussion about the grammar formalism. In Section 3.2, we show how in principle how the head-driven SCFG would translate a concrete Chinese sentence. In Section 3.3, we introduce five components of the F W S model, which would facilitate a flexible approximation to the dependencies in the head-driven SCFG. This section also serves as a mini summary for the whole thesis, as the content of the subsequent chapters discusses the development of these five components.

3.1 A Sketch of the Head-driven SCFG

As a recap, the head-driven SCFG differs from the existing SCFG in three respects: 1) the use of two nonterminal symbols, where one signifies a head and the other signifies the head's argument; 2) the modeling of the expansion of a rule as a head-

outward process, where the head is considered to be generated first followed by the head's arguments, starting from the ones closest to the head; and 3) the head-lexicalization of nonterminals, where some lexical information (the heads) in the span of the nonterminals are propagated from lower level hierarchy to the higher level one.

We develop these three distinctive features in the following SCFG rule:

$$X(h_{-L}, \ldots, h_{-1}, h_Y, h_{+1}, \ldots, h_{+R}) \rightarrow \langle \gamma, \alpha, \sim \rangle \tag{3.1}$$

$$\text{where } \gamma = X_{-L}(h_{-L}) \ldots X_{-1}(h_{-1}) Y(h_Y) X_{+1}(h_{+1}) \ldots X_{+R}(h_{+R})$$

The first difference – the use of two nonterminal symbols – is clearly reflected in the two nonterminal labels (Y and X) that appear on the rule's right hand side (RHS). The first label (Y) is a symbol for a head, which will be equated with function words to reflect the main idea of this thesis. Meanwhile, the second label (X) is a symbol for an argument of a head, which represents any span of text whose reordering is influenced by the head.

The second difference – the modeling of rule's expansion as a head-outward process – is partially reflected in the subscripts attached to the arguments (Xs), which uses the head (Y) as the point of reference. Negative indexes (-) are used for those arguments to the left of the head, while the positive indexes (+) are used for those to the right of the head. The magnitude of the index is proportional to the distance between the argument and the head with L and R indicate the total number of the left and the right arguments of the head respectively. Note that here, we overload the index not only to indicate reordering but also to indicate the arguments' position. The modeling of the head-outward process will be more articulated in one of the upcoming statistical models (named *argument selection model*), which basically uses the above indexing scheme.

The third difference – the head-lexicalization of nonterminals – is reflected

by the extra information attached to nonterminals, indicated by the h symbol inside the bracket following the nonterminal labels, which represents a set of all the heads in the span of the nonterminal. Note that at one level, there is only one active head, which is indicated by h_Y in Rule 3.1, however we design the lexicalization to propagate all the heads to provide richer information to the upcoming statistical models. As shown, all the hs are subscripted according to the position of their respective nonterminals as such they can be ordered based on their appearance on the source text.

For clarity, the target language side (α and $\sim$) is concealed; but essentially, it corresponds to one possible permutation of the source language side (γ), which actual order will be determined by one of the upcoming statistical model (named *the orientation model* and detailed later).

Additionally, the head-driven SCFG also includes the following rules:

$$X \rightarrow \langle e, f \rangle \tag{3.2}$$

$$Y(e/f) \rightarrow \langle e, f \rangle \tag{3.3}$$

$$S \rightarrow X(\bullet) \tag{3.4}$$

Rules 3.2-3.3 are terminal rules that emit the actual source (e) and target phrases (f), representing leaf nodes in the resulting hierarchical structure. The difference is that the source phrases emitted by Rule 3.3 belong to the function word class F, while those emitted by Rule 3.2 do not, splitting the entries in the phrase translation table into two disjoint sets. Note that the source and target pair in Rule 3.3 is propagated to the higher level structure as indicated by the bracket following the nonterminal on the left hand side.

Meanwhile, Rule 3.4 represents the root node in the hierarchical structure. In retrospect, this rule is similar to the glue rule in the Hiero model (Chiang, 2005) except that the reordering of Rule 3.4 is not restricted to monotone reordering. As this rule always appears at the highest level, the head-driven SCFG propagates

no information from the lower level structure where the ignored information is indicated by the • symbol (also used in Rule 3.5).

In general, the above four rules are adequate to cover most except a few exceptional cases. These exceptions include cases where there is no function word available in reordering certain span of text. To handle such an exception, the head-driven SCFG use the following *back-off* rule:

$$Y(\mathcal{U}) \quad \rightarrow \quad X(\bullet) \tag{3.5}$$

which promotes an argument to act like a head. The head-driven SCFG uses a special symbol $\mathcal{U}$ to represent such a promoted head, which will use a special statistics in the upcoming statistical model (the upcoming orientation model).

In practice, the introduction of this back-off rules unfortunately aggravates the overgeneration problem as now any phrase translation unit can become heads. We avoid this problem by making sure that this back-off rule is applicable only in cases where the first four rules are not applicable in our decoder implementation. Note that the universal token is only active at its current level and not propagated.

3.2 The Head-driven SCFG in Action

How does the head-driven SCFG translate the Chinese example in Fig. 2.1,which for browsing convenience, copied as Fig. 3.1 below?

In principle, to translate the Chinese sentence, the head-driven SCFG would need the following rules:

$$X \quad \rightarrow \quad \langle \text{表单}, \text{a form} \rangle \tag{3.6}$$

$$X \quad \rightarrow \quad \langle \text{网页}, \text{a page} \rangle \tag{3.7}$$

$$X \quad \rightarrow \quad \langle \text{数据 输入 域}, \text{data entry fields} \rangle \tag{3.8}$$

$$X \quad \rightarrow \quad \langle \text{集合}, \text{a collection} \rangle \tag{3.9}$$

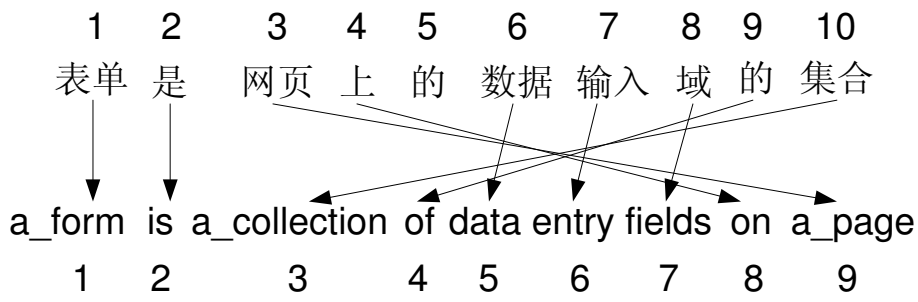

Figure 3.1: An illustration of how words move when translated, copied from Fig. 2.1.

$$Y(\text{是/is}) \rightarrow \langle \text{是}, \text{is} \rangle \tag{3.10}$$

$$Y(\text{上/on}) \rightarrow \langle \text{上}, \text{on} \rangle \tag{3.11}$$

$$Y(\text{的}_5/\epsilon) \rightarrow \langle \text{的}_5, \epsilon \rangle \tag{3.12}$$

$$Y(\text{的}_9/\text{of}) \rightarrow \langle \text{的}_9, \text{of} \rangle \tag{3.13}$$

$$X(\text{上/on}) \rightarrow \langle X_{-1}Y(\text{上/on}), Y(\text{上/on})X_{-1} \rangle \tag{3.14}$$

$$X(\text{上/on}, \text{的}_5/\epsilon) \rightarrow \langle X_{-1}(\text{上/on})Y(\text{的}_5/\epsilon)\ X_{+1}, X_{+1}\ Y(\text{的}_/\epsilon)X_{-1}(\text{上/on}) \rangle \tag{3.15}$$

$$X(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}) \rightarrow \langle X_{-1}(\text{上/on}, \text{的}_5/\epsilon)Y(\text{的}_9/\text{of})\ X_{+1}, X_{+1}Y(\text{的}_9/\text{of})X_{-1}(\text{上/on}, \text{的}_5/\epsilon) \rangle \tag{3.16}$$

$$X(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}, \text{是/is}) \rightarrow \langle X_{-1}Y(\text{ 是/is})X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}), X_{-1}Y(\text{是/is})X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}) \rangle \tag{3.17}$$

where Rules 3.6 through 3.13 are the terminal rules, and Rules 3.14 through 3.17 the nonterminal rules. Note that we attach the source word index to 的 to distinguish the two occurrence of the function word.

The head-driven SCFG translates the Chinese example by applying this set of rules in a top-down order, resulting in a derivation shown in Fig. 3.1. In particular,

the head-driven SCFG applies the rules in the following order: Rules 3.4, 3.17, 3.6, 3.10, 3.16, 3.9, 3.13, 3.15, 3.8, 3.12, 3.14, 3.7, and 3.11.

Unlike the existing SCFG with one type of nonterminal label, this grammar is better equipped to address the overgeneration problem since the lexicalization of nonterminals prevents the application of the rules in any arbitrary order. This grammar is also less susceptible to the undergeneration problem since the chance is high to learn the reorderings that occur in the nonterminal rules from the training data since they only involve high frequency words.

3.3 Architecture: Five Components

The head-driven SCFG is theoretically less susceptible to the undergeneration and the overgeneration problems, as discussed in the previous subsection. However, building such a grammar is largely non-trivial since although the head-driven SCFG only focuses on high frequency words, the number of lexical items that can be attached the nonterminals are unbounded. Thus sparse data issue may easily complicate the process, as such estimations are crucial.

The head-lexicalization itself can be implemented in many different ways. It can be hard-coded in the Xs as such each $X(h)$ would become a new type of nonterminal symbol. Or it can be treated as an attribute of X, where the value of h can vary dynamically. In this thesis, we adopt the latter since it gives us flexibility in estimating the head-driven SCFG. Note that in this case, the actual rules that are applied are the un-lexicalized version the rules.

To facilitate the approximation of the head-driven SCFG, we first break down the internal dependencies in the grammar into the following five components to be developed independently later:

1. the identity of function words (`FWID`)

$S \Rightarrow \langle X(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}, \text{是/is}),\ X(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}, \text{是/is}) \rangle$
$\Rightarrow \langle X_{-1}Y(\text{是/is})X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}), X_{-1}Y(\text{是/is})X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}) \rangle$
$\Rightarrow \langle \text{表单}\ Y(\text{是/is})X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}), \text{a form} Y(\text{是/is})X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}), \text{a form is}\ X_{+1}(\text{上/on}, \text{的}_5/\epsilon, \text{的}_9/\text{of}) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{-1}(\text{上/on}, \text{的}_5/\epsilon)Y(\text{的}_9/\text{of})\ X_{+1}, \text{a form is}\ X_{+1}Y(\text{的}_9/\text{of})X_{-1}(\text{上/on}, \text{的}_5/\epsilon) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{-1}(\text{上/on}, \text{的}_5/\epsilon)Y(\text{的}_9/\text{of})\ \text{集合}, \text{a form is a collection}\ Y(\text{的}_9/\text{of})X_{-1}(\text{上/on}, \text{的}_5/\epsilon) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{-1}(\text{上/on}, \text{的}_5/\epsilon)\text{的}_9\ \text{集合}\ , \text{a form is a collection of}\ X_{-1}(\text{上/on}, \text{的}_5/\epsilon) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{-1}(\text{上/on})Y(\text{的}_5/\epsilon)\ X_{+1}\ \text{的}_9\ \text{集合}\ , \text{a form is a collection of}\ X_{+1}\ Y(\text{的}_5/\epsilon)X_{-1}(\text{上/on}) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{-1}(\text{上/on})Y(\text{的}_5/\epsilon)\ \text{数据 输入 域 的}_9\ \text{集合}\ , \text{a form is a collection of data entry fields}\ Y(\text{的}_5/\epsilon)X_{-1}(\text{上/on}) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{-1}(\text{上/on})\text{的}_5\ \text{数据 输入 域 的}_9\ \text{集合}\ , \text{a form is a collection of data entry fields}\ X_{-1}(\text{上/on}) \rangle$
$\Rightarrow \langle \text{表单 是}\ X_{-1}Y(\text{上/on})\text{的}_5\ \text{数据 输入 域 的}_9\ \text{集合}\ , \text{a form is a collection of data entry fields}\ Y(\text{上/on})X_{-1} \rangle$
$\Rightarrow \langle \text{表单 是 网页}Y(\text{上/on})\text{的}_5\ \text{数据 输入 域 的}_9\ \text{集合}\ , \text{a form is a collection of data entry fields}\ Y(\text{上/on})\ \text{a page} \rangle$
$\Rightarrow \langle \text{表单 是 网页 上的}_5\ \text{数据 输入 域 的}_9\ \text{集合}\ , \text{a form is a collection of data entry fields on a page} \rangle$

Table 3.1: The derivation produced by the head-driven SCFG to translate the Chinese example in Fig. 3.1. The order of application of the rules is described in the text.

2. the coherence of function words' arguments (ARGCOH)

3. the selection of function words' arguments (ARGSEL)

4. the order of the application of the function words (FWORDER)

5. the bilingual orientation of function words' arguments (ARGORI)

In discussing these five components, we will relate each component with the corresponding rule it involves as well as with the three differences between the head-driven SCFG and the existing SCFGs to maintain the continuity with the previous discussion. This upcoming discussion will also serve as a mini summary for this thesis, as all the statistical models in the upcoming chapters can be seen the approximation to one of the components discussed here. Also, the labels used to refer to these components will be used frequently in the upcoming chapters. Note that the term heads and function words are heavily exchangeable in this subsection since their role in F W S model is identical.

The first component FWID is responsible for the labelling of all the terminal rules of the head-driven SCFG. In particular, it generates a function word list FW; based on which, the rule for a particular span of source text (e) is labelled, i.e. those that belong to the list $e \in F$ are labelled as heads Y (represented by Rule 3.3), otherwise $e \notin F$ are labeled as arguments X (represented by Rule 3.2). In retrospect, since this first component is related to the first distinguished characteristic of the head-driven SCFG, namely the grammar with two nonterminal labels.

Meanwhile, the second component ARGCOH is responsible for evaluating the segmentation of the arguments. In essence, the ARGCOH component's responsibility resembles a typical preprocessing step in many natural language processing tasks of identifying non-recursive and non-overlapping base phrases in the input sentence. Ideally, this component has two inter-related roles. The first role is to penalize those spans of text whose internal words would not cohere when translated. For instance,

ARGCOH should penalize a phrase translation that spans the first three Chinese words 表单 是 网页 (a form is a page) since the correct reordering requires the translation of the third Chinese word 网页 (a page) to be split from the translation of the first two words. Meanwhile, the second role is to reward those spans of text that represent maximum coherent units. Thus, for instance, ARGCOH should give a bonus score to the nonterminal that emits the following three-words phrase 数据 输入 域 (data entry fields) but none to those that just emit a one-word phrase 数据 (data) or a two-words phrase 输入 域 (entry fields). Maximum coherent units should be favored because they reduce the number of rules involved in the translation process, thus reducing the possibility of errors.

Moving on to the third component ARGSEL, this component is responsible for selecting the most appropriate set of arguments for a particular head (setting L and R parameter in Rule. 3.1), among all other possible sets. In the illustration in the previous subsection, ARGSEL correctly assigns one argument, i.e. the left neighbor, to the prepositional 上 (on) as shown in Rule 3.15 instead of the right neighbor, but two arguments, i.e. both the left and the right neighbors, to the remaining function words instead of only one argument. Note that arguments are not always positioned next to the function words – they may include non-immediate neighbors, such as the second or third neighbors. In our implementation, we develop this component as the head-outward process similar to (Collins, 2003), thus exploiting the second unique feature of the head-driven SCFG.

The fourth component, FWORDER, is responsible for assigning the order of the rule's application using the information available through lexicalization. In our example, FWORDER applies the rules in the following correct bottom-up order: the prepositional 上 (on), the first particle 的$_5$ (of), the second particle 的$_9$ and the copula 是 (are). This component exploits the third difference of the head-driven SCFG to select the correct derivation.

Up to this point the model has built the underlying hierarchical structure, but has yet to perform any reordering. This responsibility rests upon the final ARGORI component. In the example, ARGORI suggests that the arguments of the copula 是 (are) should keep their Chinese order, and that the arguments of the other function words should be translated in the inverse Chinese order. This then completes the reordering process.

Some of these components, if developed, will eventually become statistical models, including the orientation and the argument selection model which have been briefly mentioned earlier. In our implementation, we use these upcoming statistical models as features alongside seven other standard SMT features in a log-linear model, following (Och and Ney, 2002). The standard features are as follows: 1) language model $lm(e)$; 2-3) phrase translation score $\phi(e|c)$ and its inverse $\phi(c|e)$; 4-5) lexical weight $lex(e|c)$ and its inverse $lex(c|e)$; 6) word penalty wp; and 7) phrase penalty pp. We use this set of standard features as is and refer the interested readers to (Koehn, Och, and Marcu, 2003; Vogel et al., 2003) for a more elaborate discussion of these features.

The translation is then obtained from the most probable derivation of the stochastic grammar. The formula for a single derivation T is shown in Eq. (3.18), where $X_1, X_2, ..., X_{|T|}$ is a sequence of rules that involves in T with $w(X_t)$ being the weight of each particular rule X_t. $w(X_t)$ is estimated through a log-linear model, as in Eq. (3.19), where λ_j reflects the contribution of a feature f_j. The value of λ_j is obtained automatically through minimum error rate training (Och, 2003) on the development set.

$$P(T) = \prod_{t=1}^{|T|} w(X_t) \tag{3.18}$$

$$w(X_t) = \prod_{j} f_j(X_t)^{\lambda_j} \tag{3.19}$$

Throughout this thesis, we employ the standard bottom-up CKY beam parser (Cocke, 1969; Kasami, 1963; Younger, 1967) to find the target language

order which maximizes Eq. (3.18). The sketch of the decoding algorithm is discussed in Appendix A.

Chapter 4

Experimental Setup, Baselines and Pilot Study

This chapter details the data sets used, the scenarios ran and the baselines reported for all the experiments in this thesis. Here, we also describe a pilot study on the data set used to study the feasibility of our Function Words, Syntax-based (F W S) approach.

4.1 Data

In this thesis, we evaluate all experiments on a Chinese to English translation task. As our focus is on the reordering task, the standard sentence-aligned parallel corpora (traditionally used in the translation task experiments) may not be suitable to fairly evaluate the contribution of our proposal. When such parallel corpora are used, we argue that it is difficult to separate reordering-related factors from lexical-related ones. More specifically, we are unable to perform controlled experiments with *unambiguous lexical mappings* and to evaluate our proposals with respect to reordering-specific, *intrinsic evaluations*.

Having unambiguous lexical mappings is important as it removes all lexical-related problems from the decoding step, such as lexical selection ambiguity, phrase segmentation ambiguity, and out-of-vocabulary (OOV) words. Being able to perform intrinsic evaluations is also important for assessing the contribution of each proposed model to the whole reordering process. Without such evaluations, we are forced to use the standard BLEU score (Papineni et al., 2002) that evaluates our proposals with respect to the downstream translation task, in which lexical-related factors may complicate the analysis. Fortunately, we have the access to a special word-aligned parallel corpus that can leverage both unambiguous lexical mapping and intrinsic evaluations.

For all experiments, we used a corpus in the computer manual domain. Subsequently, we will refer to this corpus as the HIT corpus, since it was prepared by the Harbin Institute of Technology. We consider this HIT corpus special because it comes with manual word alignment, which refers to word-level correspondences between the source and target sentences assigned manually by human annotators. To the best of our knowledge, the HIT corpus represents the largest manually word-aligned corpus available to the research community as of this thesis writing. Table 4.1 shows a snapshot of one sentence pair with its annotation from the corpus, which has been used as our running example.

Following the standard open-test setup, we divided this corpus into three sets: the training set, the development set and the testing set. We randomly assigned the sentence pairs of each set, except that we forbid the sentences longer than 30 words to be assigned to the development and the testing sets. In our experiments, we used the training and the development sets to estimate the parameters and the weighting factor of each proposed model, respectively. We evaluated the performance of each proposed model on the testing set and reported the figure as the final evaluation result. Table 4.2 shows the statistics of each individual set.

a/1 form/2 is/3 a/4 collection/5 of/6 data/7 entry/8 fields/9 on/10 a/11 page/12
表单/1 是/2 网页/3 上/4 的/5 数据/6 输入/7 域/8 的/9 集合/10
(2:1); (3:2); (5:10); (6:9); (7:6); (8:7); (9:8); (10:4); (12:3);

Table 4.1: A snapshot of HIT corpus. The first line refers to the English sentence, the second line to the corresponding Chinese sentence, while the third line to the word alignment. The word alignment takes the format of $(i{:}j)$ where i refers to the position of the English word while j to the position of the aligned Chinese word.

Note that the size of the testing and development sets is almost identical to the standard corpora, although the size of the training set is smaller. Of course, we expect to train the statistical models on a larger set of training set but we think that this corpus is adequate for our purpose since the parameter size of our models, as we will show later, is independent of the corpus size and we only focus on a small set of very frequent words. The size of the training data is arguably also appropriate for the baselines model (described shortly) since the vocabulary size of our corpus is relatively modest (around 4,000 words).

	Number of	Number of words	
	sentence pairs	Chinese	English
training (train)	7,000	145,731	135,032
development (dev)	1,000	13,986	14,638
testing (test)	2,000	27,732	28,490

Table 4.2: Statistics of the HIT corpus.

4.1.1 Gold Standard Function Words

In addition to the manually word-aligned corpus, we also obtained a list of genuine Chinese function words, which is hereafter referred to as the **gold standard function words**. We asked a linguist to manually extract this gold standard list from (Howard, 2002), which contains over 1,000 regularly used Chinese function words.

Throughout the thesis, we use this list extensively for experiments and evaluations. In particular, we use this list in the upcoming pilot study for assessing the feasibility of the idea of using function words for reordering. Furthermore, we use this list to measure the benefit of having a genuine list of function words on the reordering quality in Chapter 6. Finally, we use this list to evaluate the performance of our upcoming approximation to the `FWORDER` component in Chapter 8.

4.2 Two Scenarios: Perfect Lexical Choice and Full Translation Task

Although we focus on the reordering task, we are also interested in evaluating our proposals on the real translation task where the F W S approach has to deal with lexical-related ambiguities and noisy word alignment. Thus, we devise two scenarios: *perfect lexical choice* and *full translation task*, where the first scenario reflects our focus on the reordering task while the second one reflects our interest on the translation task.

In the perfect lexical choice scenario, the task is to rearrange the target sentence which is originally translated in the source language order into the target language order. In the context of the Chinese to English translation, the task is to recover the correct order of the English sentence from the scrambled Chinese order. In this scenario, we fully utilize the manual word alignment available in the HIT corpus in training the model parameters. To ensure the absence of lexical-related

ambiguities and out-of-vocabulary problem, we construct the phrase translation table (which would become the terminal rules, i.e. Rules 3.2-3.3 in the head-driven SCFG) from the alignment available in the testing set at the individual word level, such that each word in the test set has exactly one possible lexical mapping – the correct one. Note that in this scenario, we create the phrase translation table for the development set in the same way. The absence of lexical-related ambiguities also suggests that all the standard phrase-based features are turned off during decoding time. We want to emphasize that even though it seems that we use the testing set for the construction of the phrase translation table, the final evaluation results still reflect a valid open test, since we train all other pertinent models entirely on the training set.

Moving to the full translation task scenario, the task is more complex as our proposed F W S approach has to deal with lexical-related ambiguities, representing a real world translation task. Taking only the source sentence as input, the F W S approach not only has to reorder the sentence into the target language order but also has to find the appropriate translation for each source word. In this scenario, we ignore the manual word alignment and rely on the automatically-obtained one in training all the models parameters including the phrase translation table.

To automatically construct the word alignment, the standard procedure typically adopted by other phrase-based models is run. First, the automatic word aligner GIZA++ (Och and Ney, 2003) is used to extract two uni-directional word alignments over the training data: one from Chinese to English and the other from English to Chinese. The two alignments are post-processed using the "grow-diag-final-and" heuristic (Koehn et al., 2005) to form a symmetrical bi-directional alignment. All the proposed models including the phrase translation table are all trained on this symmetrized alignment.

Unlike the first scenario, each word in the testing set is now potentially

subject to out-of-vocabulary, lexical mapping as well as segmentation ambiguity problems. That is, one Chinese word may or may not have an English translation; it can belong to many different segmentations and each segmentation can have more than one possible lexical mapping.

These ambiguities complicate not only the reordering process but also the evaluation and especially the error analyses. Specifically, they make it difficult to pinpoint the exact cause of the the performance increase (or drop), simply because too many factors are involved. To work around this issue, we follow Chan et al. (2007). The idea is to use the intermediate results (p_+, p_-, p_0) produced by the sign-test comparing a system against a baseline. The p_+ refers to the sentences where the system performs better than the baseline, p_- refers to the sentences where the system performs worse than the baseline, while the p_0 refers to the sentences where the system and the baseline perform equally well.

To perform the sign-test, we follow (Collins, Koehn, and Kucerova, 2005). Specifically, we start by calculating the BLEU score for the baseline system and continue by substituting one sentence in the baseline with the corresponding sentence in the system output. We classify the sentence into the p_+, p_- or p_0 if the BLEU score of the new set is better than, worse than or equal to the BLEU score of the baseline. We do this procedure for every sentence in the testing set, by keeping all other sentences the same. For analysis, we look at the sentences in p_+ and p_- to assess whether the changes we propose affect the performance positively or negatively. As such, we consider our proposal contributes positively if the changes appear more in p_+ than in p_-. Although stronger analysis is needed to make a more rigorous conclusion, we consider such analysis provides an adequate indication about the positive (or negative) contribution of our proposal.

For all experiments, we used the publicly available SRILM-Toolkit (Stolcke, 2002) in its default setting to train a trigram language model over the English side

of the training data. We also ran David Chiang's implementation of the minimum error rate training procedure (Och, 2003) over the development set to estimate the weighting factor, i.e. λ in Eq. 3.19.

4.3 Baselines

In order to meaningfully evaluate our proposed models, it is useful to have baseline systems to situate the evaluation results. We describe the baseline systems below and report their performance in the upcoming appropriate chapters. To facilitate a fair comparison, we define the standard settings which are used consistently not only by the baseline systems but also by our proposed system. The shared settings are as follows: 1) the maximum beam size = 100; 2) the maximum number of words in a phrase translation unit (also in a hierarchical phrase translation) = 5; and 3) the bi-directional alignment heuristic = "grow-diag-final-and".

4.3.1 Pharaoh

Pharaoh (Koehn, 2004a) represents the first state-of-the-art phrase-based system. This system employs the distortion penalty model as its reordering model (Koehn, Och, and Marcu, 2003), taking the following penalty-based formula:

$$d(a, b) = e^{|a_i - b_{i-1} - 1|} \tag{4.1}$$

where a and b are the current and the previous translated English phrases, respectively; while a_i is the start position of a in the Chinese sentence and b_{i-1} is the end position of b in the Chinese sentence. This model basically penalizes those non-monotone reorderings that digress from monotone reordering. Although simple, this system performs comparably well in many translation competitions (Koehn and Monz, 2005; Koehn and Monz, 2006). For the full translation task scenario,

we use the off-the-shelf decoder, while for the perfect lexical choice scenario, we faithfully integrate the distortion penalty model into our decoder. Also, in the experimental section of Chapter 5, we use Eq. 4.1 as an evaluation metric and report the log value of d of the whole testing set as `dist`. We use this metric to indicate the aggressiveness of a system in reordering the input sentences, where high value indicates an aggressive reordering.

4.3.2 Moses

Moses (Koehn et al., 2007) is a direct replacement of Pharaoh, representing the current state-of-the-art phrase-based model. This system incorporates a more advanced reordering model, which pays attention to the lexical identity of the phrase being moved (a), similar to the unigram lexicalized reordering model (Tillman, 2004). In particular, the reordering model of Moses takes the following form:

$$P(orientation|lex(a)) \tag{4.2}$$

where $lex(a)$ is the lexical identity of the phrase being moved and *orientation* is one of these three orientation values: monotone, swap and discontinuous, which are analogous respectively to the Left, Right and Neutral values, described in Section 2.2. Note that we can only fairly produce the performance of this baseline system in the full translation task scenario, since in the perfect lexical choice scenario, the definition of a and b is fixed at the word level, which makes the extraction of the orientation value less reliable.

4.3.3 Hiero

The Hiero model (Chiang, 2005) represents the state-of-the-art syntax-based system. It has performed significantly better than the phrase-based system and comparably better than other syntax-based systems (Chiang, 2007). Hiero represents

a strong baseline for syntax-based models as the rank (the number of nonterminals on the RHS) of Hiero rules can extend effectively to five if lexical items are considered as (pseudo) nonterminals, which is higher than the rank of any other formally syntax-based models, including our proposed approach. Following the original setting, we specify the maximum length of the Hiero's initial phrases to 12 and maximum number of lexical items in the sub-phrases to 5. Note that although desirable, the performance for this model cannot be extracted for the perfect lexical choice scenario because the phrase translation table contains only single word mappings, from which no hierarchical rules can be extracted.

4.4 A Pilot Study

Here, we want to assess the feasibility of our proposed function word idea on the HIT corpus. We seek to do so by examining how often function words are involved in non-monotone reorderings. We concentrate only on these cases since only in such cases, phrases need to be reordered.

To facilitate this pilot study, we first develop simple approximations to the `FWID` and the `ARGSEL` components. As a reminder, the `FWID` component is responsible for generating a function words list, based on which a phrase translation unit is labelled; while the `ARGSEL` component is responsible for assigning the appropriate arguments for a certain function word. In particular, we introduce the most-frequent heuristic as the approximation to the `FWID` component which equates function words to the N most frequent words; and the immediate-neighbor heuristic as the approximation to the `ARGSEL` component which assumes that a function word only influences their immediate arguments (thus L and R in Rule 3.1). In this pilot study, we take full advantage of the manual word-alignment available in the HIT corpus so that the non-monotone reorderings measured resemble true phenomena in real languages.

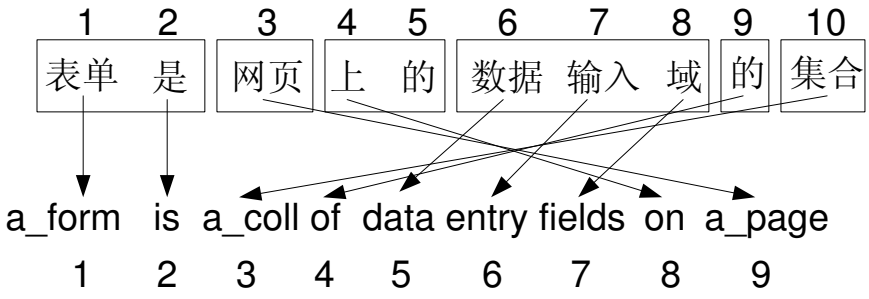

Figure 4.1: The running example that is partitioned into a sequence of max-mono phrase translations. A max-mono phrase translation is indicated by one rectangular box.

We start this pilot study by first segmenting the sentence pair into a sequence of *maximum monotone* phrase translations (*max-mono* in short), which refer to those phrase translations which internal word alignments are all monotone and cannot be merged with any other phrase translations without violating the all-monotone constraint. Fig. 4.1 illustrates how the running example is segmented into a series of max-mono phrase translations. Note that we attach unaligned words consistently to the left phrase translation when possible. In this study, we consider two consecutive max-mono phrase translations as one case of non-monotone reordering.

In total, there are 6,244 non-monotone reorderings in the testing set. We consider a function word involved in a case of non-monotone reordering only if the bordering words contain the function word. We define bordering words as follows: suppose a is the current max-mono phrase translation and b is the preceding max-mono phrase translation, then the bordering words are the union of those words that range from b's last aligned word to b's last word and those words that range from a's first word to a's first aligned word. For instance, the bordering words of the third 上 的 (on) and the fourth 数据 输入 域 (data entry fields) max-mono phrase translations are 上 (on), 的 (a Chinese particle) and 数据 (data).

N	#involvement	%	avg phrase length when not involved	avg phrase length when involved
1	2,017	32.30	5.23	5.91
4	3,727	59.69	5.09	5.7
16	4,706	75.37	4.55	5.75
64	5,610	89.85	3.68	5.65
128	5,942	95.16	3.14	5.57
256	6,115	97.93	2.92	5.5
gold (318)	5,387	86.28	4.07	5.67
1,024	6,232	99.81	2.08	5.45
all (2,352)	6,244	100	-	5.45

Table 4.3: Statistics of non-monotone reordering cases where function words are involved.

Table 4.3 shows the statistics of non-monotone reorderings that are influenced by function words. In this pilot study, we consider two types of function words: 1) function words that are obtained from the gold standard list; and 2) function words that are obtained from the most-frequent heuristic with different cut-off value N – thus function words are the top N most frequent words in the corpus.

As shown in Table 4.3, the number of non-monotone cases involving function words is very high. If the gold standard function words are used, function words are involved in more than 86% of cases. The proportion is also high when the function words used are estimated from simple most-frequent heuristic. Some of the function words that are involved in non-monotone reordering include the following function words: 为 (for), 的 (of), 到 (to) and 在 (at), which are also involved in the transformational rules defined in (Wang, Collins, and Koehn, 2007).

As for the remaining cases, a closer inspection reveals that non-monotone cases which do not involve function words mostly consist of base noun phrase constructions or adverb-verb constructions. For instance, no function word involves in

the translation of the following Chinese noun phrase which consists of two words: 图表 (chart) 类型 (type,kind) to "a kind of chart". Similarly, no function word involves in the translation of 自动 (automatically) 启动 (start) to "start automatically".

In such cases, we appeal to the strength of the phrase-based approach since, as shown in Table 4.3, the average length of a and b combined is less than the maximum phrase length we set for our experiments. Nevertheless, the high proportion of non-monotone reordering cases which involve function words strongly supports our idea of using function words as the basis to address the reordering task, confirming the feasibility of the proposed F W S approach.

Chapter 5

The Basic F W S Model

In this chapter, we introduce the basic Function Word, Syntax-based (F W S) model, which serves as a natural starting point for the development of our function word reordering idea to the syntax-based framework. In developing this basic model, we essentially want to demonstrate the potential of the F W S approach by developing some simple approximations for all the five components of the F W S approach described in Chapter 3 and evaluate its performance through intrinsic and extrinsic evaluations.

The outline of this chapter is as follows. We first describe the exact grammar formalism for the basic F W S model in Section 5.1, influenced by the use of the `immediate-neighbor` heuristic as the approximation to the `ARGSEL` component. We then develop the approximations to `ARGORI`, `FWORDER` and `ARGCOH` components in Section 5.2. We revisit the roles of these components when we discuss their approximation.

As for the approximations to the `FWID` component which main responsibility is to identify function words, we reuse the `most-frequent` heuristic. Thus, function words are equated with the top N most frequent words in the corpus. Like the `immediate-neighbor` heuristic, the `most-frequent` heuristic have been described

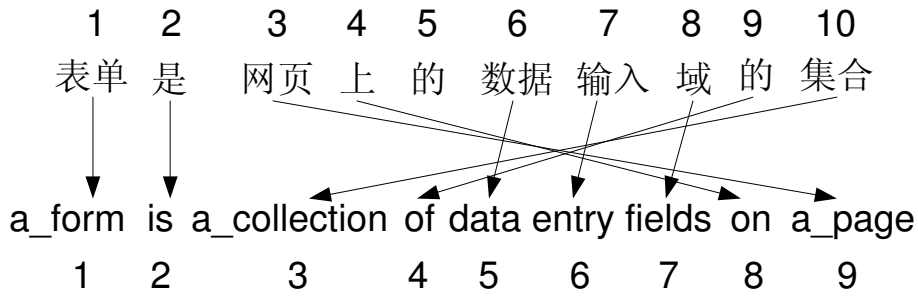

Figure 5.1: An illustration of how words move when translated.

and used in Section 4.4 about the pilot study. Subsequently, we discuss the parameter estimation method in Section 5.3. We report experimental results in Section 5.4. Finally, we end with error analysis of the result and a discussion in Section 5.5.

For illustrations, we use the same running example as in the previous chapters, which we copy here for browsing convenience as Fig. 5.1.

5.1 The Grammar

Here, we discuss the head-driven SCFG rewrite rules used by the basic F W S model. The head-driven SCFG rules used by the basic F W S model are shaped by the `immediate-neighbor` heuristic which is the model's approximation to the `ARGSEL` component. Recall that this component is responsible for assigning the appropriate arguments for a particular function word. This heuristic, introduced in Chapter 4, specifies that the arguments of a head can only be the head's immediate neighbors, i.e. its left or right neighbors.

This `immediate-neighbor` heuristic sets the parameters L and R of Rule 3 to be at most 1, constraining the influence of Y only to either or both X_{-_1} and X_{+_1}.

Given this constraint, the non-terminal rules of the head-driven SCFG consists of:

$$X(h_{-1}, h_Y, h_{+1}) \rightarrow \langle X_{-1}(h_{-1})\ Y(h_Y)\ X_{+1}(h_{+1}), \alpha, \sim \rangle \quad (5.1)$$

$$X(h_{-1}, h_Y) \rightarrow \langle X_{-1}(h_{-1})\ Y(h_Y), \alpha, \sim \rangle \quad (5.2)$$

$$X(h_Y, h_{+1}) \rightarrow \langle Y(h_Y)\ X_{+1}(h_{+1}), \alpha, \sim \rangle \quad (5.3)$$

$$X(h_Y) \rightarrow \langle Y(h_Y), \alpha, \sim \rangle \quad (5.4)$$

where the subscripts of the arguments indicate the arguments' position on the source side with respect to the head (Y) as described earlier in Section 3.1. In short, the positive (+) and negative (-) signs indicate that the arguments are on the left and the right of Y respectively, while the number indicates the distance between the arguments and the head.

As shown, this heuristic allows four different rewrite rules. Rule 5.1 models cases where the function word would influence the reordering of both its left and right arguments. Meanwhile, Rules 5.2 and 5.3 model cases where the function word only influences one argument, i.e. the left and the right one respectively. Finally, Rule 5.4 models cases where the function word doesn't influence any argument, which useful in cases where there are two competing function words appear next to each other thus one has to become the argument of the other.

In addition to Rules 5.1-5.4, the basic F W S model also uses all other head-driven SCFG rules, i.e. Rules 3.2-3.4, as well as the back-off rule i.e. Rule 3.5, that are sketched in Chapter 3. The $\alpha, \sim$ pair in the rules represents the target language ordering, which will be determined by our upcoming approximation to the `ARGORI` component.

Note that our decoder is a CKY-style decoder, which requires all the rules to have the rank at most two. Since the rank of Rule 5.1 is three, we have to binarize the rule into several intermediate rules. Since the intermediate binarized rules can be reduced to either Rule 5.3-5.2, we reuse the above rules and attach an extra

information to indicate whether the rules are the final rule with rank two or the intermediate rules to be merged to form rules of rank three. Appendix A provides a more detail description of the decoding algorithm, including how to emulate rules of rank higher than two. In retrospect, the basic F W S model uses the Bracketing Transduction Grammar (BTG) similar to the Bruin model ((Xiong, Liu, and Lin, 2006; Deyi Xiong and Lin, 2008)), in the sense that both in essence consists the BTG's straight and inverted rules.

5.2 Statistical Models

In this section, we develop the statistical models for the `ARGORI`, `FWORDER` and `ARGCOH` components. As a recap, the `ARGORI` component is responsible for assigning the target language ordering $(\alpha, \sim)$. the `FWORDER` component is responsible for deciding the order of rule's application, and the `ARGCOH` is responsible for rewarding or penalizing a certain span of text based on whether it will be translated coherently or not. We will start from the development of the `ARGORI`, `FWORDER` and finally `ARGCOH` components. These three resulting models will be come three separate features in the log-linear formula described in Eq. 3.19.

5.2.1 Orientation Model

We call our approximation to the `ARGORI` component, which responsibility is to assign the target language order of source phrases, as the *orientation model (ori)*. In this model, we put our function word idea into practice by first developing `pORI` function. Specifically, we define `pORI` as a function that takes two inputs – a function word and its argument – and outputs the argument's new location relative to the function word's position.

For the output, we adopt orientation values similar to those in (Nagata et

al., 2006), with the exception that due to a different decoding process, the values here refer to the orientation in the target sentence. As a case in point, we use X_{+1} (the first neighbor to the right of Y) in the following discussion. But, this definition is generalizable to other arguments at other locations since the orientation value is symmetric, i.e. the same value still holds even if the positions of the function word and the argument are swapped.

Formally, the `pORI` function takes the following form:

$$\texttt{pORI}(Y, X_{+1}) = o, \qquad \text{where } o \in \{\text{MA,RA,MG,RG}\} \tag{5.5}$$

mapping Y and X_{+1} into one of four different orientation values:

- Monotone-Adjacent (MA): Y and X_{+1} are in the same order as the source side and there is *no* intervening phrase between them.
- Reverse-Adjacent (RA): Y and X_{+1} are in *inverse* source order and there is *no* intervening phrase between them.
- Monotone-Gap (MG): Y and X_{+1} are in the same order as the source side but there is an intervening phrase between them.
- Reverse-Gap (RG): Y and X_{+1} are in *inverse* source order but there is an intervening phrase between them.

Basically, the four orientation values are the combination of *directionality* (i.e. Monotone or Reverse) and *adjacency* aspect (i.e. Adjacent or Gapped). The directionality aspect refers to whether the function word and its argument maintain the source language order, while the adjacency aspect refers to the presence (or the absence) of an intervening phrase between the function word and its argument in the target language. Fig. 5.2 illustrates the four orientation values.

Table 5.1 shows the distribution of the `pORI` values for the some of the most frequent words in the HIT corpus, including those words that are involved in the

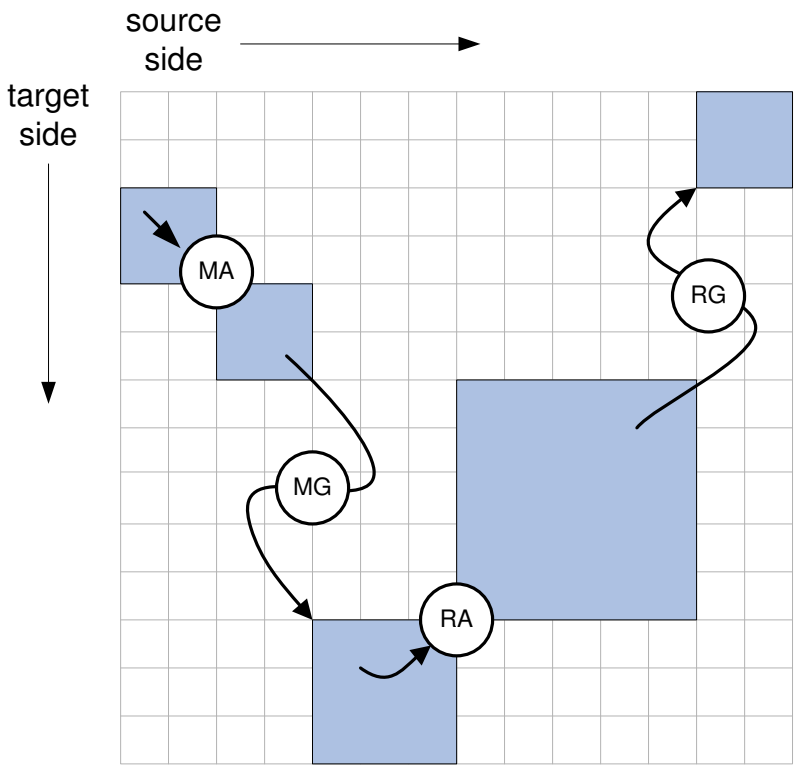

Figure 5.2: An alignment matrix to illustrate the four orientation values, defined in the text. Each gray box represents a phrase translation.

Rank	Word	X_{-1}				X_{+1}			
		MA	RA	MG	RG	MA	RA	MG	RG
1	的	0.45	**0.52**	0.01	0.02	0.44	**0.52**	0.01	0.03
2	,	**0.85**	0.12	0.02	0.01	**0.84**	0.12	0.02	0.02
3	。	**0.99**	0.01	0.00	0.00	**0.92**	0.08	0.00	0.00
4	”	**0.87**	0.10	0.02	0.00	**0.82**	0.12	0.05	0.02
5	“	**0.84**	0.11	0.01	0.04	**0.88**	0.11	0.01	0.01
6	和	**0.95**	0.02	0.01	0.01	**0.97**	0.02	0.01	0.00
7	任务	**0.73**	0.12	0.10	0.04	**0.51**	0.14	0.14	0.20
8	可以	**0.78**	0.12	0.03	0.07	**0.86**	0.05	0.08	0.01
9	或	**0.95**	0.02	0.02	0.01	**0.96**	0.01	0.02	0.01
10	将	**0.87**	0.10	0.01	0.02	**0.88**	0.10	0.01	0.00
21	是	**0.85**	0.11	0.02	0.02	**0.85**	0.04	0.09	0.02
37	上	0.33	**0.65**	0.02	0.01	0.31	**0.63**	0.03	0.03
-	$\mathcal{U}$	**0.76**	0.14	0.06	0.05	**0.74**	0.13	0.07	0.06

Table 5.1: Orientation statistics of selected frequent Chinese words in the HIT corpus. $\mathcal{U}$ denotes the universal token. Dominant orientations of each word are in **bold**. The list is ranked according to the token's unigram probability.

running example. We will describe the exact method to compute these statistics in the subsequent section but discuss the statistics here. To some extent, these statistics reflect our linguistic intuition about the syntactic difference that may be encoded in function words. For example, the orientation statistics for 是 (to be) overwhelmingly suggest that the grammar should preserve the Chinese order when translating the arguments of the copula, reflecting the fact that the copula has the same role in both languages, i.e. joining the left and the right noun phrases. Meanwhile, the orientation statistics for the word 上 (on) suggest that the grammar should reorder the argument in the inverse Chinese order, reflecting the shift from Chinese postposition construction to the English preposition one. Similarly, the dominant orientation for the particle 的 (of) is equal to the noun-phrase shift from modifier-modified to modified-modifier, which is common when translating Chinese noun phrases into English.

Table 5.1 also includes a special token ($\mathcal{U}$), which will be subsequently referred to as the universal token. Recall that this universal token is the token

propagated by the head-driven SCFG when it promotes an argument to take the role of a head, as modelled by Rule 3.5. We design the statistics of this token to capture the orientation statistics at aggregate level, representing the tendency of a word in the source language in reordering its neighboring phrases to a certain orientation when translated to the target language. As shown in Table 5.1, the universal token's statistics strongly suggest that the English sentence should preserve the Chinese language order most of the time – a similar preference as the one reported by (Nagata et al., 2006). For our approach, this information is invaluable, particularly in cases where no function word is involved and some reordering decisions must be made.

Once the `pORI` function is defined, the development of the orientation model is straightforward. Taking Rule 5.1 as a case in point, we define the orientation model (*ori*) of that rule as:

$$ori(X(h_{-1}, h_Y, h_{+1}) \rightarrow \langle X_{-1}(h_{-1})\ Y(h_Y)\ X_{+1}(h_{+1}), \alpha, \sim \rangle) =$$

$$P(\texttt{pORI}(X_{-1}, Y)|Y, \texttt{pORI}(X_{+1}, Y)) \times P(\texttt{pORI}(X_{+1}, Y)|Y, \texttt{pORI}(X_{-1}, Y)) \tag{5.6}$$

where the orientation model score for Rule 5.1 consists of two factors: the probability of X_{-1}'s orientation given X_{+1}'s orientation and the probability of X_{+1}'s orientation given X_{-1}'s orientation. Conditioning the model on the other argument's orientation is necessary to prevent the orientation model from allocating probability mass to already occupied locations. The orientation model score for Rules 5.2 and 5.3 share the same basic principle except that since these rules only have one argument, its orientation model score only depends the probability of its argument's orientation.

5.2.2 Preference Model

We develop the *preference model (pref)* as an approximation to the `FWORDER` component. Given two rules, the primary responsibility of this model is to arbitrate which rule should take precedence, i.e. to have a higher position in the hierarchical structure. The preference model performs the arbitration in a simple manner by looking at the frequency information of the heads of these two rules. More concretely, this model gives precedence to higher frequency words, ensuring that they always have the maximum number of arguments.

The intuition behind this model is that more frequent words have more reliable statistics than less frequent ones, thus they should be given priority to reorder more arguments. Taking Rule 5.1 as a case in point, we approximate its preference model score as:

$$pref(X(h_{-1}, h_Y, h_{+1}) \rightarrow \langle X_{-1}(h_{-1})\ Y(h_Y)\ X_{+1}(h_{+1}), \alpha, \sim \rangle) = uni(h_Y) \qquad (5.7)$$

where *uni* is a function that outputs the unigram probability of a token. The preference model score for all other rules are similar.

5.2.3 Phrase Boundary Model

We develop the *phrase boundary model (pb)* as a simple approximation to the `ARGCOH` component. The responsibility of this model is to check whether a terminal rule emits a coherent argument, i.e. the internal words stay or move together. In general, the definition of coherent argument depends on many linguistic-related factors, such as whether the arguments have the same syntactic category across the two languages.

In this basic model, we propose a simple approximation by employing a shallow linguistic analysis via a text chunker. The idea is that coherent arguments tend to occupy spans of text that observe the syntactic boundary of the source language.

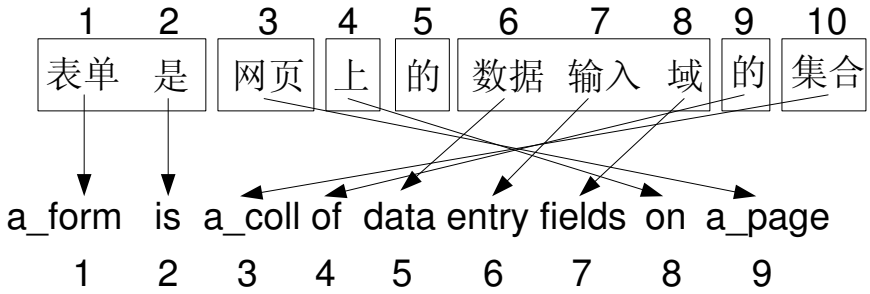

Figure 5.3: The running example which is annotated with syntactic boundary information. A syntactic phrase is illustrated as a sequence of Chinese words in a rectangular box.

Fig. 5.3 shows the running example annotated with chunking information.

We develop this phrase boundary model as a penalty-based model, soft constraining the phrase translations to conform the source constituent boundary. The *pb* model only applies to terminal rules (Rule 3.2) and takes the following form:

$$pb(X \rightarrow \langle e, f \rangle) = \begin{cases} 0 & \text{if the rule emits a syntactic Chinese phrase} \\ -1 & \text{otherwise} \end{cases} \tag{5.8}$$

Note that in this model we relax the knowledge-poor assumption as we are only seeking for a simple approximation but we intend to seek a knowledge-poor solution in the future.

5.3 Parameter Estimation

This section focuses only on the extraction of the terminal rules of the head-driven SCFG and the parameter estimation for the orientation and the preference models, since the parameters for the phrase boundary model can be estimated directly from the output of a standard text chunker. In our experiments, we use (Chen, Zhang, and Isahara, 2006). We train the orientation and preference models from the

statistics of the training data by first deriving the event counts and then computing the relative frequency for each event.

Since the nonterminal rules are pre-defined, we only need to extract the terminal rules (i.e. Rules 3.2-3.3) from parallel data. To do so, we use the standard method employed by the phrase-based models, which relies on the consistent alignment heuristic. The detail of the heuristic has been discussed in Chapter 2 and copied below for browsing convenience.

$$\mathcal{PT}(f_1^J, e_1^I, A) = (f_j^{j+jj}, e_i^{i+ii}) : \forall (i', j') \in A : j \leq j' \leq j + jj \leftrightarrow i \leq i' \leq i + ii \tag{5.9}$$

where $\mathcal{PT}$ stands for phrase translations, f_1^J and e_1^I are the source and target sentences of length J and I respectively, A is a set of alignments (i', j') between f_1^J and e_1^I and i and j are used to indicate source and target word indexes respectively. The consistent alignment heuristic basically specifies that a source phrase (f_j^{j+jj}) of length jj and its translation e_i^{i+ii} of length ii is a valid phrase translation if the source phrase is only aligned with the words inside its translation. For the perfect lexical choice scenario, the length of the source phrase (jj) is limited to 1, while in the full translation task scenario, it is limited to a certain predefined number.

The parameter estimation for the orientation model involves harvesting statistics of $(f/e, o)$ tuples for each source and target translation pair f/e where $o \in$ {MA, RA, MG, RG} is the orientation value of an argument. We pair f with its translation e in the hope that such a pairing would capture the different role f may have. For instance, 的 can act either as a noun phrase or as a prepositional marker. Apparently, the translation of 的 would be different in each case. More concretely, it translates to "of" if it acts as a noun phrase marker just as 的 at position 5 in the running example, or it translates to nothing if it acts as a prepositional phrase just as 的 at position 9 in the running example. Additionally, we restrict the definition of f only to word level to alleviate data sparsity concern. The distribution

in Table 5.1 is computed by marginalizing f over its all possible translations.

These tuples unfortunately are not directly observable in parallel corpora. Thus here, we develop an algorithm to estimate the unseen events of $(f/e, o)$. To refer to the counts of the unseen events, we use the term *soft count* to refer to the counts of unseen events that are obtained via a heuristic; as opposed to the hard count that is computable only if the events (in this case, the annotation about arguments) are observable. Note that for the basic F W S model, we must extract two $(f/e, o)$s: one for the left and one for the right argument; however, we omit references to them since both left and right statistics share identical training steps. In fact, the same procedure is generalizable to all other arguments at other locations.

As input, the algorithm expects parallel corpora with word-to-word alignments, obtained from either manual annotation or an automatic process. Then, given an enumeration of all words in the corpora, it hypothesizes the left (X_{-_1}) and the right (X_{+_1}) arguments of each f/e. This is done by using a heuristic called **Maximum Consistent Alignment** (MCA), which is exactly the same as the consistent alignment heuristic (Och and Ney, 2004) traditionally used to construct the phrase translation table, except with the additional "maximum" condition. We add the "maximum" condition since we are only interested in the largest consistent phrase translations, as such each f/e has exactly one unique argument to its left and to its right.

Additionally, the "maximum" condition helps to prevent overestimating the gapped orientation (**MG** or **RG**) is not incorrectly suggested, which is important to prevent many false non-monotone reorderings. Fig. 5.4 illustrates the case where a loose definition of argument will lead to a different orientation value. Suppose we want to extract the $(f/e, o)$ for the left argument of the last function word 的 (of), then defining only the neighboring word 域 (fields) as f/e's left argument would

result in the gapped orientation as illustrated in Fig. 5.4b with the phrase 数据输入 (data entry) considered as a gap. In contrast, the MCA heuristic correctly suggests the desired adjacent orientation since it considers the whole neighboring phrase 数据输入域 (data entry fields) as f/e's left argument, as illustrated in Fig. 5.4a.

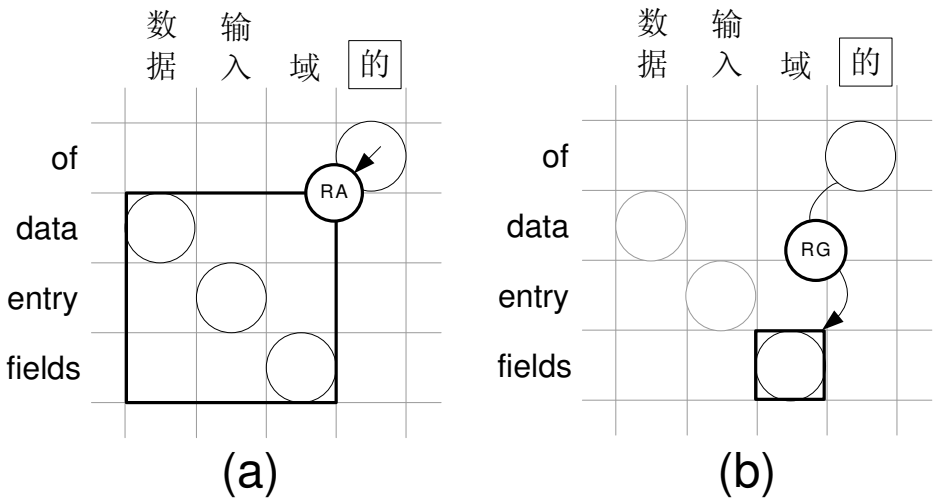

Figure 5.4: Illustrations of the correctly learnt (part a) and the incorrectly learnt (part b) arguments of the function word 的(of). The arguments are indicated by the thickly outlined rectangular. The correct orientation, which is RA, is suggested if the MCA (the box in part a) is used. The incorrect orientation, which is RG, is suggested if only the immediate neighboring word (the box in part b) is used.

Once the arguments are estimated, the o value can be directly extracted by inspecting the directionality (Monotone or Reverse) and the adjacency aspects (Adjacent or Gapped) of the arguments with respect to their corresponding f/e. Concretely, both the head and its argument are phrase translations. Formally, suppose the head is $f_{j_1}^{j_2}/e_{i_1}^{i_2}$ while the argument is $f_{j_3}^{j_4}/e_{i_3}^{i_4}$, then in terms of directionality, the argument's orientation is monotone if $i_2 < i_3$ and reverse if $i_1 < i_4$, while in terms of the adjacency, the argument's orientation is adjacent if $(|i_3 - i_2| == 1 \vee |i_1 - i_4| == 1)$ and gapped if $(|i_3 - i_2| \neq 1 \wedge |i_1 - i_4| \neq 1)$.

We record the occurrences of each particular $(f/e, o)$ as their soft counts $C(f/e, o)$. Once all f/es have been enumerated and their corresponding soft counts $C(f/e, o)$s are available, we can estimate the orientation model for a particular f/e and the universal token $\mathcal{U}$ using the maximum likelihood principle as follows:

$$P(o|f/e) = C(f/e, o)/C(f/e, \cdot), Rank(e) \leq N \quad (5.10)$$

$$P(o|\mathcal{U}) = \sum_{Rank(e)>N} C(f/e, o) / \sum_{Rank(e)>N} C(f/e, \cdot) \quad (5.11)$$

Samples of these statistics are in Table 5.1 and applicable to the running example.

Meanwhile, the parameter estimation for the preference model is simple, since the event of interest is directly observable. Given the unigram counts $C(e)$, we estimate the preference model for f/e and $\mathcal{U}$ as follows:

$$uni(e) = C(e)/C(\cdot), Rank(e) \leq N \quad (5.12)$$

$$uni(\mathcal{U}) = 1/(|V| - N) \sum_{Rank(e)>N} C(e)/C(\cdot) \quad (5.13)$$

where $|V|$ indicates the vocabulary size and $Rank$ is a function that outputs the rank of a word based on its unigram probability. Note that in estimating the preference model, we are only interested in the source language side f of the head.

5.4 Experiments

In inquiring the potential of the F W S approach, we performed experiments with these three specific purposes: 1) to study how well we approximate the `ARGORI` component, 2) to study how our approximation affects the reordering quality, and 3) to evaluate the performance of the system in the full translation task. To achieve this purpose, we evaluated the basic F W S model against the Pharaoh system[1]

[1]We provide the performance of the stronger baselines in the later chapters, since here we only probe the feasibility of the F W S model.

using intrinsic and extrinsic evaluations, namely `pORI-acc` and BLEU respectively. The `pORI-acc` evaluates the basic F W S with regard to how well the model approximates the `pORI` of the function words' left and right arguments, while the BLEU score evaluates the basic F W S with regard to how well the translation output matches a reference translation. Here, we report the `pORI-acc` as the aggregate for all the words in the corpus and the BLEU score as the case insensitive BLEU-4. Besides these two metrics, we also used `dist`, which we mention in Section 4.3.1, to indicate how aggressive a system is in reordering the input sentences, i.e. the higher the value the more aggressive the system is. For `pORI-acc` and `dist` metrics, manual word alignment is essential. We use the methods described in Section 4.2 to construct the phrase translation tables for these two scenarios. Note that entries in the phrase translation unit serve as terminal rules (Rules 3.2-3.3) in the head-driven SCFG.

5.4.1 Perfect Lexical Choice

Here, the task is to recover the correct order of the English sentence from the scrambled Chinese order, free from lexical-related ambiguities. We fully utilized the manual word alignment provided by the HIT corpus to train the model parameters.

Table 5.2 compares `pORI-acc` and BLEU between the basic F W S model and the baseline. As shown, we report several baseline models, which are all in $N = 0$ column. The first baseline (*mono*) represents a system that employs the distortion penalty model only, preferring monotone reordering; while the second baseline (d) represents a system that emulates the Pharaoh system, coupling together the language model and the distortion penalty model. The third baseline ($ori, N = 0$) represents a system that relies only on the language model component, which is equivalent to our basic F W S with no active head. From this model, we study the behavior of the F W S model with different numbers of heads N. To identify

	N=	0	1	4	16	64	128	256	1,024
pORI-acc	*mono*	66.39							
	d	73.52							
	ori	64.66	76.40	76.59	77.35	77.94	**78.89**	79.53	*79.63*
	ori+pref		76.34	76.69	77.28	77.89	**78.45**	78.99	*78.96*
	ori+pref+pb		76.33	76.74	77.34	77.82	**78.43**	78.87	*78.96*
BLEU	*mono*	68.88							
	d	76.46							
	ori	68.39	77.68	77.78	78.44	79.00	**79.58**	80.11	*80.07*
	ori+pref		77.77	78.23	78.65	79.41	**79.69**	80.07	*80.17*
	ori+pref+pb		77.77	78.28	78.67	79.46	**79.78**	79.99	*80.24*

Table 5.2: Results using manual word alignment input. Here, the baselines are in the $N = 0$ column; *ori*, *ori+pref* and *ori+pref+pb* are different F W S configurations. The results of the model (where N is varied) that features the largest gain are in **bold**, whereas the highest score is *italicized*.

the heads, we apply the `most-frequent` heuristic, developed in Chapter 4, which equates the top N most frequent words as heads. Starting with the language model alone (N=0), we incrementally add the orientation (ori), preference ($ori + pref$) and phrase boundary models ($ori + pref + pb$).

As shown in Table 5.2, the lowest performing system is the third baseline ($ori, N = 0$) which relies only on the language model component. A closer inspection on the translation output suggests that the language model component tends to recommend non-monotone reorderings aggressively . Such a tendency hurts the performance, since in the reference, the majority of reorderings (66.39%) are monotone reordering as indicated by `pORI-acc` of the *mono* system. Thus, including a distortion penalty model that discourages non-monotone reorderings increases the accuracy to 73.52% as shown in row *d*. The `dist` value in Table 5.3 gives the same insight, indicated by the `dist` value of the third baseline which is much lower than the ground truth value. Table 5.3 also shows that incorporating the distortion penalty model curbs the aggressivity of the language model.

N=	0	1	4	16	64	128	256	1,024
mono	0							
d	11,790							
ori	35,182	19,238	18,928	20,752	21,868	23,214	23,784	23,988
ori+pref		20,166	20,556	21,104	20,816	21,632	21,270	20,826
ori+pref+pb		19,980	20,208	20,778	20,636	21,242	21,078	20,564
ground truth								**31,789**

Table 5.3: The `dist` value of all the systems reported in Table 5.2. The ground truth is also reported in the last row in **bold**.

When we incorporate the orientation model, we can see improvements even by just modeling the most frequent word (的). This model promotes non-monotone reordering conservatively only around the function word (where the dominant statistic suggests reverse ordering), while promoting monotone reordering in all other cases. As shown, increasing the value of N leads to greater improvements. Among these experiments, we obtain the most effective improvement by setting N to 128. We can obtain additional but marginal improvements by increasing N further. The highest improvement can be obtained at the expense of modeling an additional 900+ lexical items. Similarly, this trend is also observed for the BLEU score.

Lastly, we study the effect of the preference ($pref$) and the phrase boundary (pb) models. Apparently, the inclusion of both statistical models has little effect on the orientation accuracy, although it improves BLEU consistently – but by only a small margin. These results suggest that perhaps although both models correct the mistakes made by the orientation model, they make new errors. We will provide more detailed error analyses in the last section.

5.4.2 Full SMT experiments

Here, we train all the models on noisy, automatically-obtained word alignment. We employed the same baseline systems as the ones in the perfect lexical scenario. Note

that for the second baseline, we employed the Pharaoh's own decoder. Since the Pharaoh decoder restricts long-distance reordering, we ran the minimum error rate training for different distortion limits from 0 to 10 for a fair comparison and only report the best parameter (dl=5).

For F W S model, we use the phrase translation table similar to the baseline system and run an identical set of experiments as the perfect lexical choice scenario, except that we report only the result for N=128 as this value gives the most effective improvement. Table 5.4 reports the performance in BLEU scores.

The same trend similar to the perfect lexical choice is also observed here, where the language model is too aggressive in recommending non-monotone reorderings and coupling the language model with the distortion penalty model improves the BLEU score.

More importantly, the same trend of improvement is also shown by the basic F W S model over the baseline systems. In particular, the basic F W S model improves the BLEU score over the baseline and the improvement is statistically significant at $p < 0.01$. We also observe the same trend as the one in the perfect lexical choice scenario for the preference and the phrase boundary models. The fact that the phrase boundary model yields no noticeable improvement is similar to the previous findings reported in (Chiang, 2005; Koehn, Och, and Marcu, 2003). Nevertheless, this set of experiments shows that the simple F W S approach can perform well even in the experiments with lexical-related ambiguities present.

Table 5.4 also shows the `dist` value of the systems. As shown, the `dist` value of the Pharaoh system is much lower than the basic F W S model, suggesting the Pharaoh's bias toward monotone reordering. Note that the `dist` values here are not comparable with the ones in the perfect lexical choice since variable-length phrase translations are used and even may not be comparable to the other values in the same table. Nevertheless, this value indicates that Pharaoh does not move phrases

System	BLEU	dist
mono	21.51	0
$ori, N = 0$	21.40	43,174
Pharaoh (dl=5)	22.44	7,010
ori	24.92	18,408
$ori + pref$	25.06	18,304
$ori + pref + pb$	25.11	17,078

Table 5.4: Results for the full translation task scenario.

as much as it should. To some extent, this is confirmed by our casual inspection on the Pharaoh output which reveals that some of the reordering mistakes made by the Pharaoh system are due to its inability to accommodate the long-distance reordering phenomena. This is partly due to the hard restriction imposed by the distortion limit parameter but we suspect it is more due to the distortion penalty model that discourages non-monotone reorderings.

5.5 Discussion

In this section, we provide some in-depth error analysis on the experimental results to understand the strengths and weaknesses of the basic F W S model. We are particularly interested in analyzing the output produced by the basic F W S which parameter gives the most efficient improvement, i.e. $N = 128$. While such basic model is able to correctly assign the `pORI` predicate in 78.89% of cases, it apparently fails to assign the correct `pORI` value in 21.11% other cases. Here, we focus on analyzing these 21.11% cases. The discussion in this section will eventually motivate the development of the subsequent improved F W S model. In our discussion, we try to relate the error as much as possible to the five components of the F W S approach.

Table 5.5 visualizes in a matrix form, the discrepancy between the prediction

made by the basic F W S model and the ground truth extracted from the manual alignment. Based on this table and some casual inspections, we discuss our analyses in the following subsections. The first three subsections discuss the three errors which will be addressed in the subsequent three chapters, while the last subsection discusses one other error which we reserve for future work. Note that we always relate these errors to the components of the F W S approach.

5.5.1 Error 1: the number of heads that support non-monotone reordering is too few

The overly conservative monotone reordering is as detrimental as the overly aggressive non-monotone reordering. The `dist` value of the basic F W S model, which is much higher than the ground truth, indicates that the basic F W S model is still very conservative in suggesting the non-monotone reordering. Table 5.5 provides an insight that most of the mistakes are due to the model's failure to predict non-monotone reorderings; 77.5% to be more precise (considering all the columns except the first column). Among these cases, the majority is due to the basic F W S model's strong tendency to suggest monotone reordering, which constitutes 57% of cases (the total of the first row).

We find one possible reason behind such a strong tendency toward monotone reordering when we inspect the orientation statistics of the words in the head list. As indicated in Table 5.1, the orientation of most heads strongly prefer monotone reordering. Among all the heads that support monotone reordering, we find that most of them are content words, such as 任务 (task) which ranks 7^{th} in Table 5.1. We suspect that one possible reason behind the overly strong tendency toward monotone reordering is because there is not enough function words that support non-monotone reorderings in the top N most frequent words.

Thus, we hypothesize that we can improve the performance further by im-

				Ground Truth													
				MA **MA** (1)	MA MG (2)	**MA** **RG** (3)	**RA** **RA** (4)	**RA** **MG** (5)	RA RG (6)	MG MA (7)	**MG** **RA** (8)	MG MG (9)	MG RG (10)	**RG** **MA** (11)	RG RA (12)	RG MG (13)	RG RG (14)	Total
Prediction	**MA**	**MA**	(1)	×	390	420	725	323	15	530	299	22	41	393	46	39	2	3,245
	MA	MG	(2)	139	×	60	2	42	1	2	15	4	2	3	1	3	0	274
	MA	**RG**	(3)	181	14	×	29	9	8	5	27	1	6	26	1	4	5	316
	RA	**RA**	(4)	351	8	13	×	11	6	1	7	0	0	5	40	2	0	444
	RA	**MG**	(5)	124	48	13	39	×	29	9	36	3	10	57	1	15	1	385
	RA	RG	(6)	0	0	4	6	1	×	0	0	0	1	0	0	0	0	12
	MG	MA	(7)	223	5	5	6	24	0	×	29	2	10	53	2	7	1	367
	MG	**RA**	(8)	136	5	67	61	14	8	21	×	1	15	17	5	8	1	359
	RG	**MA**	(9)	120	1	26	17	31	3	34	11	0	2	×	5	7	1	258
	RG	RA	(10)	2	0	1	2	0	0	0	1	0	0	1	×	1	0	8
Total				1,276	471	609	887	455	70	602	425	33	87	555	101	86	11	5,668

Table 5.5: The matrix that shows the discrepancy between the prediction made by the F W S model and the ground truth extracted from the manual word alignment. The headers contain three pieces of information: the orientation for the left argument, the orientation for the right argument and the (column/row) index. The headers in **bold** indicates the orientation values that can be accommodated by the basic F W S model.

proving the approximation of the FWID component. We hope that a better approximation of the FWID component can identify genuine function words that provide stronger evidences toward non-monotone reordering. We detail our new proposal for the the FWID component in Chapter 6.

To evaluate the upcoming proposal, we introduce a new metric, which we call false-mono. This value refers to the number of cases where a system falsely assigns monotone reordering, obtained by summing the first row of Table 5.5. The false-mono value for the basic F W S model is 3,245. The goal is thus to reduce the number of false-mono error.

5.5.2 Error 2: the type of arguments handled by the heads is too limited

In total, there are 16 possible pairs of orientation value for the left and right arguments of a head with 14 of which are observed as shown in Table 5.5. However, there are only 6 possible pairs of orientation values that can be accommodated by the basic F W S model. We refer to these 6 cases as *handled cases* while the other 8 cases as *unhandled cases*. The basic F W S model is essentially an SCFG which can only emit contiguous phrases on the source and target sides, while on the other hand, some of the unhandled cases correspond to target phrases that are non-contiguous. For example, the basic F W S model cannot modelled the orientation values MA and MG because it is not capable to emit a gap between the head and the right arguments on the target language side. We illustrate the six cases of handled argument in Fig. 5.5 and highlight them in Table 5.5 by presenting their header in bold style.

Apparently, the total number of unhandled argument cases is quite significant. If we consider the union of the rows and the columns of the unhandled cases (headers not bolded), it makes up around 36.69% of the total mistakes of the

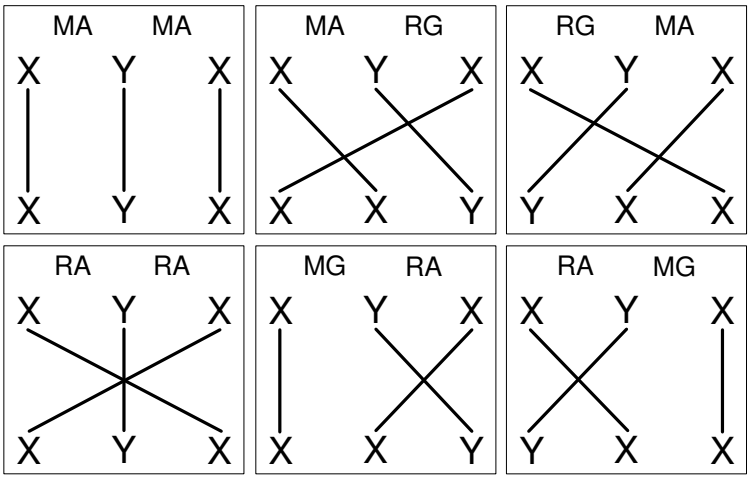

Figure 5.5: Six combinations of orientation values that can be accommodated by the basic F W S.

model. In these cases, the model is at the mercy of the language model or other heads to correctly position the unhandled arguments to the correct position. This analysis argues for a better approximation to the `ARGSEL` component, since allowing more flexible set of arguments (e.g. the second neighbor argument to the left or the right) would allow the F W S model to accommodate the unhandled argument cases. We will give more concrete illustrations and propose a new approximation in Chapter 7.

To evaluate the upcoming new approximation, we introduce a new evaluation metric, which we call the `unhandled-arg`. The `unhandled-arg` counts the number of errors that is attributed to the arguments unhandled by the `immediate-neighbor` heuristic. The value for the basic F W S model is 2,080, obtained by counting the union of rows and columns which headers are not bolded. The goal is thus to reduce the number of `unhandled-arg`.

5.5.3 Error 3: the estimation of the FWORDER component is too weak

The basic F W S model develops the preference ($pref$) model as the approximation to the FWORDER component. This model hypothesizes that more frequent words should influence the reordering of more arguments than less frequent ones, thus appear higher level in the hierarchical structure. However, the experimental result shows that this model is only able to give marginal improvement over the baseline F W S model without the preference model.

When we analyze the results, we observe the following. Although there are some cases where it is beneficial to have the more frequent words to influence more arguments, there are also some cases where it is detrimental. Fig. 5.6 illustrates such a case.

In Fig. 5.6, there are four heads involved: 任务 (task), 分配 (assign), 资源 (resources) and 时 (when); which ranks are 7, 83, 16 and 69 respectively. Out of these four heads, only the dominant orientation of the fourth function words 时 (when) is non-monotone. Arranging the ordering of the heads by unigram statistics results in Fig. 5.6b, where 时 (when) is not allowed to take arguments because its rank is one of the lowest.

A better approximation is clearly needed since such errors are quite common. A conclusion can be drawn from the inaccuracies of the preference models: the unigram formulation of the FWORDER component is too weak to suggest the correct level a head should appear. A better formulation should include more contextual information, perhaps by incorporating the competing word, i.e. the head word of the arguments, into the model. We detail our new approximation to the FWORDER component and a new intrinsic evaluation metric in Chapter 8.

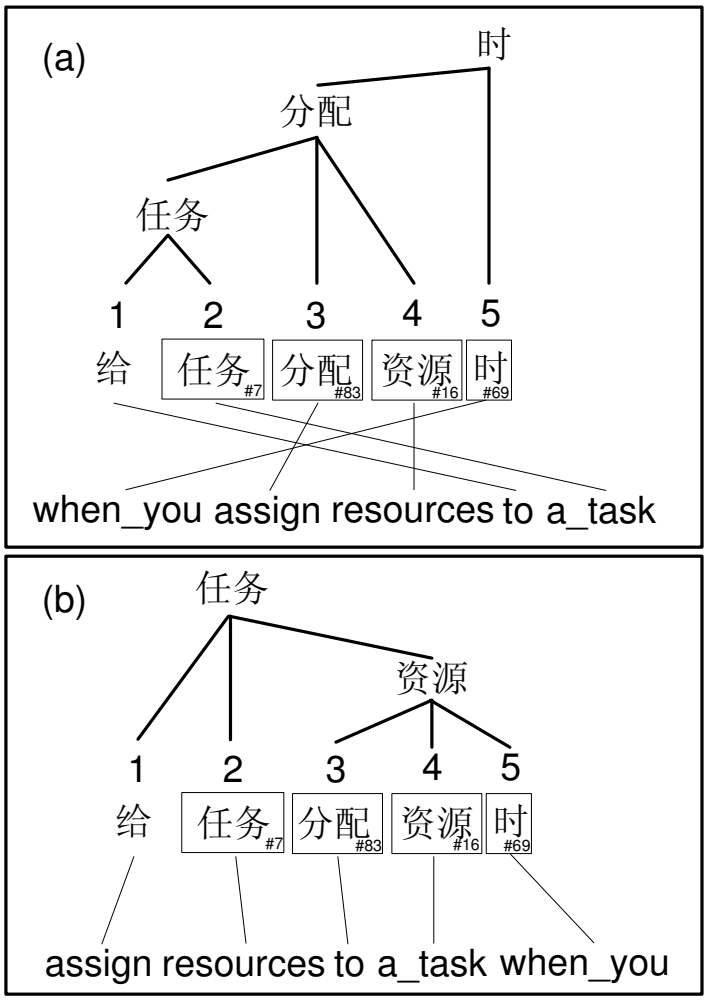

Figure 5.6: An illustration where the preference model fails to produce the correct vertical ordering of function words. The heads are Chinese characters in the box and their ranks are indicated by the number in the box. The node's label indicates the head that is currently active reordering its arguments at that level. (a) represents the correct vertical ordering as a reference. (b) represents the wrong vertical ordering where the vertical ordering of heads is arranged by the ranks of the heads.

5.5.4 One other error

Here, we discuss and analyze another type of error that concerns with the weak approximation of the ARGCOH component. Recall that the ARGCOH component has two roles, i.e. 1) to penalize those phrase units that do not cohere when translated; and 2) to reward those phrase translation units that are in the maximum sense. The basic F W S model develops the phrase boundary (*pb*) model to approximate the ARGCOH component. This model equates the coherence of a phrase translation with whether the phrase translation observes the source syntactic boundary. Apparently, the result shows that this model only produces marginal improvement over the baseline model which does not employ the *pb* model. To understand the underlying cause, we analyzed the chunking information and compared it with the max-mono phrase translations which we used in the pilot study in Chapter 4.

In total, the text chunker partitions the test sentences into 21,636 segments, from initially 27,332 words. When we verified these segments, we found out that their quality is relatively good, i.e. violating the maximum-monotone constraint only in 313 cases. However, the number of segments partitioned by the text chunker is still too large if we compare it with the number of max-mono phrase translations which is 6,244. This suggests that the *pb* model may fail to perform its second role of grouping the words into a maximum coherent unit.

We performed a small experiment to understand the potential of having perfect information about argument coherence. To do so, we utilized the max-mono phrase translations but break them into smaller segments if they contain function words. For instance, if a max-mono phrase translation consists of the following words "$f_1 f_2 f_3 f_4$" where only f_3 is a function word, then we break this unit into three phrase translations: "$f_1 f_2$", "f_3" and "f_4". This procedure results in 23,959 number of segments. For this small experiment, we use the same exact setting as $ori, N = 128$, differing only in the phrase translation table. Note that

unlike the *pb* model, here, we are imposing the segmentation information. When we run this experiment, we can achieve 81.10 BLEU point or 1.50 BLEU point above the baseline model. This result suggests that it is important to find a better approximation to the `ARGCOH` model, perhaps by incorporating more sophisticated linguistic information. We will return to this point again in the last chapter when we discuss the future work.

Chapter 6

Function Word Identification

This chapter concerns with improving the approximation of the `FWID` component, which responsibility is to identify a list of function words which would become the heads in the head-driven SCFG. Identifying heads represents the first step in the F W S model, which needs to be accurate to ensure the success of downstream processes. In Chapter 4, we introduce the simple `most-frequent` heuristic which equates the top N most frequent words in the corpus as function words. This simple heuristic is also used by the basic F W S model in Chapter 5, allowing the model to achieve a relative good reordering quality on a Chinese to English reordering task.

However, our error analysis on the output of the basic F W S model in Section 5.5.1 shows that one of the basic F W S model's systematic error is due to the weakness of this simple heuristic. To improve the function word identification process, here, we propose a new heuristic called the `deviate-frequent`, which use the so-called deviation statistics (detailed shortly) to complement the frequency statistics used in the `most-frequent` heuristic. Note that since this heuristic still has no access to linguistic annotation, the identified function words may not all necessarily genuine, however, we hope that these words are more suitable for the reordering purpose.

6.1 Motivation

Error analysis in Section 5.5.1 suggests that one of the most prominent mistakes made by the basic F W S model concerns with the model's failure to correctly recommend non-monotone reordering. Our casual observation suggests that this mistake correlates with the number of heads in the function word list that recommend non-monotone reordering. This is evident in the orientation statistics of the function words list created by the `most-frequent` heuristic that mostly support monotone reordering. Thus, our hope here is to generate a list that contains as many function words that capture non-monotone reorderings as possible. We develop this intuition into a new heuristic called the `deviate-frequent` heuristic.

Essentially, the `deviate-frequent` heuristic combines two statistics – the frequency and the deviation statistics – that will be used to test whether a word should belong to a function word class or not. The frequency statistics measure how many times a word appears in the corpus and have been used by the `most-frequent` heuristics. Meanwhile, the deviation statistics measures how different the orientation statistics of a word are from the orientation statistics of the universal token $\mathcal{U}$.

The idea behind the `most-frequent` statistics comes from our simple observation that the orientation statistics of content words are quite similar from those of the universal token. For instance, Table 5.1 shows that the orientation statistics of the content word 任务 (task) strongly suggest MA (monotone adjacent) orientation with roughly the same distribution as the orientation statistics of the universal token. In this regard, modeling content words is redundant since if the words were not modeled, the same reordering would still be suggested anyway.

We still keep the frequency statistics since we want to maintain the high level of coverage over the data. The frequency statistics can also compensate the adverse effects caused by unreliable deviation statistics as due to the low count,

low frequency words tend to vary more from the universal token's orientation. We observe in our initial experiments that considering only the deviation statistic may unfairly assign more weight to low frequency words, which can hurt the reordering task. In summary, the words identified by the `deviate-frequent` heuristic are those words that appear frequently and have non-trivial orientation statistics.

6.2 Ranking Words with Frequency and Deviation Statistics

In this section, we first describe the method to estimate the deviation statistic and then proceed to the complete description of the `deviate-frequent` heuristic. The estimation of the frequency statistics can be done in a straightforward manner by simple word counting, thus omitted.

Let the orientation vector of a word f be defined as follows:

$$\overrightarrow{ori_f} = \begin{bmatrix} ori(\texttt{pORI}(X_{-1}, f) = \text{MA}|f) \\ ori(\texttt{pORI}(X_{-1}, f) = \text{RA}|f) \\ ori(\texttt{pORI}(X_{-1}, f) = \text{MG}|f) \\ ori(\texttt{pORI}(X_{-1}, f) = \text{RG}|f) \\ ori(\texttt{pORI}(X_{+1}, f) = \text{MA}|f) \\ ori(\texttt{pORI}(X_{+1}, f) = \text{RA}|f) \\ ori(\texttt{pORI}(X_{+1}, f) = \text{MG}|f) \\ ori(\texttt{pORI}(X_{+1}, f) = \text{RG}|f) \end{bmatrix} \tag{6.1}$$

where the elements are taken from the orientation model's parameters. Note that here, we are looking at the orientation statistics of an individual word at coarse level, marginalized f over all of its possible translations.

Let the same vector be defined for the universal token $\mathcal{U}$ as follows:

$$\overrightarrow{ori_{\mathcal{U}}} = \begin{bmatrix} ori(\texttt{pORI}(X_{-1},\mathcal{U}) = \text{MA}|\mathcal{U}) \\ ori(\texttt{pORI}(X_{-1},\mathcal{U}) = \text{RA}|\mathcal{U}) \\ ori(\texttt{pORI}(X_{-1},\mathcal{U}) = \text{MG}|\mathcal{U}) \\ ori(\texttt{pORI}(X_{-1},\mathcal{U}) = \text{RG}|\mathcal{U}) \\ ori(\texttt{pORI}(X_{+1},\mathcal{U}) = \text{MA}|\mathcal{U}) \\ ori(\texttt{pORI}(X_{+1},\mathcal{U}) = \text{RA}|\mathcal{U}) \\ ori(\texttt{pORI}(X_{+1},\mathcal{U}) = \text{MG}|\mathcal{U}) \\ ori(\texttt{pORI}(X_{+1},\mathcal{U}) = \text{RG}|\mathcal{U}) \end{bmatrix} \tag{6.2}$$

Using these two vectors, we define the deviation statistic dev_f as the following root mean square deviation (RMSD) formula:

$$dev_f = \sqrt{\frac{\sum_{|\overrightarrow{ori}|}^{k=1}(\overrightarrow{ori_f}[k] - \overrightarrow{ori_{\mathcal{U}}}[k])^2}{|\overrightarrow{ori}|}} \tag{6.3}$$

The denominator is constant for all words, thus can be safely ignore in practice.

Before combining, we normalize the deviation and the frequency statistics:

$$devnorm_f = \frac{dev_f - min(\forall'_f dev_{f'})}{max(\forall_{f'} dev_{f'}) - min(\forall_{f'} dev_{f'})} \tag{6.4}$$

$$freqnorm_f = \frac{log(uni(f)) - min(\forall_{f'} log(uni(f')))}{max(\forall_{f'} log(uni(f'))) - min(\forall_{f'} log(uni(f')))} \tag{6.5}$$

where $uni(f)$ is the unigram probability of a word, which has been introduced for the preference model in the basic F W S model.

The final figure df_f is obtained from the linear combination of the two statistics using δ to control the contribution of each statistic:

$$df_f = \delta.freqnorm_f + (1-\delta).devnorm_f \tag{6.6}$$

where $\delta = 1$ brings us back to the `most-frequent` heuristic while $\delta = 0$ makes the identification process relies entirely on the deviation statistics. We determine the appropriate value for δ empirically.

The function word identification process ends by sorting all words according to its df_f score and equating the top N best words as function words. Note that we transform the frequency statistic to its log form because of the facts that we combine the two statistics in a linear fashion and that the underlying distribution of the frequency statistic is not linear but exponential.

6.3 Experiments

Here, we study the effect of modeling head words obtained from different heuristics. The purpose of this section is as follows: 1) to evaluate the performance of the `deviate-frequent` heuristic with respect to the reordering quality; 2) to validate whether our proposal to remedy the basic F W S model's first systematic error is effective; and 3) to verify whether the success (or the failure) of our proposal extends to the full translation task.

Before pursuing the above goals, we first establish the performance of using the gold standard function word identities in Section 6.3.1 where we used the gold standard function word list, described in Section 4.1.1. Then in Section 6.3.2, we report our efforts of pursuing the first and the second purposes in the perfect lexical choice scenario. To evaluate the impact of our proposal on the basic F W S model first error, we used the `false-mono` metric discussed in Section 5.5.1, which counts how many times the model falsely predict non-monotone reordering for monotone reordering. Thus, the lower is the better. To evaluate the reordering quality, we use the standard BLEU score. Finally in Section 6.3.3, we report our effort to pursue the third goal in the full translation task scenario.

6.3.1 Gold Standard Function Words

In this set of experiments, we wanted to establish the performance of having perfect knowledge, where the model can correctly identifies all genuine function words. To do so, we used the gold standard function word list, described in Section 4.1.1. In total, there are 318 words in the testing set that belong to the gold standard function word list. These words constitute 59.6% of all the words in the testing set. For our first baseline, we used the top 318 most frequent words, representing a model which has the same number of lexical items as in the gold standard function word list. For the second baseline, we truncated this list to the top 152 most frequent words, representing a model which roughly covers the same amount of words in test set as the genuine function words do. The experiments reported here share an identical setup as the *ori* setting in the basic F W S model, differing only in the heads modeled.

System	BLEU	Coverage
$ori, FW = gold$	78.19	59.64
$ori, N = 318$	80.32	77.23
$ori, N = 152$	79.75	59.87

Table 6.1: Results of using the gold standard function word inventory versus using those obtained from the `most-frequent` heuristic. The third column (Coverage) refers to the words coverage over the testing set

We report the results in Table 6.1, where $ori, FW = gold$ refers to the experiments using the gold standard function word list, $ori, N = 318$ to the first baseline and $ori, N = 152$ to the second baseline. As shown, using genuine function words apparently performs worse than the two baselines. This result runs counter with our intuition that using the gold standard function words should result in an improvement.

Inspecting the results, we uncovered a couple of causes. First, there are

some gold standard function words that appear only a few times in the training data. Apparently, the orientation statistics of these genuine but low frequency function words are not reliable, thus causing incorrect reorderings. For instance, among the 318 function words modeled, only 120 words appear more than 5 times in the corpus. In contrast, the lexical items modeled by the baseline model have more reliable statistics since they always appear in high frequency. Second, the gold standard function word list is still not as exhaustive as we hope. Unfortunately, the missing function words include some important function words like the preposition 中 (in) which strongly support non-monotone reordering. This is perhaps due to the fact that the distinction between function words and content words is often vague. For instance, the word 中 (in) is possibly considered as a verb (to hit) by (Howard, 2002); from which the list was extracted. Regardless of the results, this set of experiments give an insight that having reliable statistics is vital.

6.3.2 Perfect Lexical Choice

Here, we study the effect of the proposed `deviate-frequent` heuristic on the reordering task. Specifically, we study the effect of different value of δ in terms of the BLEU score. We report the results in Table 6.2, which also includes the statistics about the list's coverage over the testing data and its intersection with the list produced by original frequency-based heuristic as well as with the gold standard function words.

In experiments reported in Table 6.2, the same number of function words is used ($N = 128$). $\delta = 1.0$ represents the baseline where only the frequency statistic is used, while $\delta = 0.0$ represents the performance where only the deviation statistic is used. Respectively, inters1 and inters2 represent the number of words shared with the words obtained by the `most-frequent` heuristic and with the gold standard function words respectively. `false-mono` reports the number of errors attributed

$\delta =$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
`false-mono`	3,068	3,087	3,029	3,057	3,067	3,052	3,048	3,030	3,057	3,092	3,245
(in %)	54.59	55.36	54.18	54.87	54.95	55.06	55.17	55.13	55.36	55.46	57.25
BLEU	*79.45*	79.68	79.92	79.95	79.91	**79.97**	79.84	79.89	79.89	79.64	*79.58*
cov. (%) s	38.22	44.85	50.50	53.94	54.99	56.58	57.31	57.48	57.91	57.90	58.13
inters1	61	69	77	84	93	102	110	115	121	124	128
inters2	39	42	45	49	50	50	47	49	49	49	46

Table 6.2: Results of using the `deviate-frequent` heuristic, reported over different δ value. The baseline is in *italics* while the best result is in **bold**.

to the false monotone reordering. Our goal is to reduce this value.

We can see some encouraging results in Table 6.2. Modeling the function words identified by the `deviate-frequent` heuristic reduces the number of `false-mono` errors as the contribution of the deviation statistics increases. The same trend is also observed in terms of BLEU score, where the reduction of the `false-mono` error leads to the increase in BLEU score. This trend continues up to a certain value $\delta = 0.5$, where the number of function words in the list is the highest among all other values.

When we manually inspected the list of head words produced at $\delta = 0.5$, we found better quality heads. Table 6.3 shows some samples of added and removed heads of that setting. As shown, the added words include some genuine function words, such as: 由 (from), 后 (positional marker), 但(but), 还,(yet), 内(within) and 每 (each), which have some tendencies toward non-monotone reordering; while the removed words mostly include nouns, verbs and adjectives, which statistics are relatively similar to the universal token's statistics. In summary, this set of experiments shows that our new approximation to the `FWID` component corrects some of the basic F W S model's error of falsely recommending monotone reordering.

Chi-nese	English	X_{-1}				X_{+1}			
		MA	RA	MG	RG	MA	RA	MG	RG
Universal Tokens									
$\mathcal{U}$	-	**0.76**	0.14	0.06	0.05	**0.74**	0.13	0.07	0.06
Removed Heads									
管理	supervise	**79.70**	7.43	4.46	8.42	**83.66**	2.97	8.42	4.95
支持	support	**81.70**	8.09	7.23	2.98	**74.89**	7.23	11.06	6.81
邮件	mail	**78.98**	10.80	5.68	4.55	**55.68**	12.50	15.91	15.91
系统	system	**71.81**	8.39	11.74	8.05	**68.46**	9.40	11.74	10.40
无法	unable	**76.05**	3.59	4.79	15.57	**94.01**	4.79	1.20	0.00
需要	require	**74.19**	9.68	11.61	4.52	**79.68**	11.94	4.52	3.87
更改	change	**72.73**	9.09	10.30	7.88	**68.48**	4.24	19.39	7.88
帮助	assist	**85.63**	5.99	4.19	4.19	**80.24**	0.60	13.17	5.99
连接	connect	**71.30**	16.09	6.09	6.52	**62.17**	6.09	19.57	12.17
自动	automatic	**57.67**	8.99	14.29	19.05	**77.78**	17.99	3.70	0.53
Added Heads									
由	from	39.29	5.95	**52.38**	2.38	**89.29**	5.95	2.38	2.38
参阅	consult	**95.56**	0.00	1.48	2.96	**97.78**	0.74	1.48	0.00
章	section	**68.63**	31.37	0.00	0.00	3.92	7.84	26.47	**61.76**
后	after	**45.74**	44.96	3.10	6.20	**44.96**	43.41	10.08	1.55
框	frame	**84.78**	3.26	11.96	0.00	17.39	5.43	**71.74**	5.43
只	only	**54.17**	5.83	38.33	1.67	**61.67**	33.33	1.67	3.33
但	but	**94.56**	2.72	1.36	1.36	**94.56**	4.08	0.68	0.68
还	yet	**60.99**	15.60	21.99	1.42	**60.28**	35.46	2.84	1.42
内	with	30.77	**64.84**	0.00	4.40	29.67	**65.93**	3.30	1.10
每	each	**48.76**	3.31	40.50	7.44	**90.08**	4.13	3.31	2.48

Table 6.3: Samples of some removed words that are no longer considered and some added words that are newly considered as heads by δ=0.5 as compared to δ=1.0. The dominant orientation of each head's arguments is in **bold**.

System	BLEU
ori, δ=1.0	24.92
ori, δ=0.5	25.29

Table 6.4: BLEU scores for the full translation task scenario. $ori, \delta = 1.0$ represents the baseline taken from Chapter 5 where the head identification only involves the frequency statistics, $ori, \delta = 0.5$ represents the system that combines the frequency and deviation statistics with equal weight.

6.3.3 Full Translation Task

Here, we want to verify whether the same performance improvement in the previous scenario also applies in the full translation task, where the deviation statistics are calculated from the noisy orientation statistics. In particular, we compared the translation performance of the basic F W S system using the `most-frequent` heuristic versus the F W S system using the `deviate-frequent` heuristic. For the proposed `deviate-frequent` heuristic, we used $\delta = 0.5$, which produced the best reordering quality in the perfect lexical choice scenario. Table 6.4 reports the full translation task experiments. As shown, employing the `deviate-frequent` heuristic improves the performance - although not statistically significant.

To further study the result, we analyzed the intermediate results of the statistical significance test. In particular, we were interested in examining whether our proposed approximation makes more changes in p_+ (where *ori,δ=0.5* performs better than *ori,δ=1.0*) or in p_- (where *ori,δ=0.5* performs worse than *ori,δ=1.0*).

Table 6.5 shows the statistics of the testing sentences classified into the three sets. We further analyzed the sentences by focusing on those sentences that contain both the added and the removed heads. We assume that the performance of these sentences would best represent the effect of having a different set of heads. In total, there are 275 of such sentences, out of the 2,000 sentence pairs as indicated in the intersection column. Although the sample size is relatively small, it is enough to

Set	Sign test	intersection	
	Count	Count	%
$\mathbf{p_+}$	637	124	45.10
p_0	743	53	19.27
p_-	620	98	35.63
Total	2,000	275	100

Table 6.5: The comparison between *ori,δ=0.5* and *ori,δ=1.0*. p_+ refers to *ori,δ=0.5*>*ori,δ=1.0*; p_- refers to *ori,δ=0.5*<*ori,δ=1.0*, while p_0 refers to *ori,δ=0.5* = *ori,δ=1.0*. The column labeled "intersection" refers to the number of sentences in each set which source sentence contains both the added heads and the removed heads. Between p_+ and p_-, the one with more sentences is in in **bold**.

indicate the effect of employing the `deviate-frequent` statistics. As shown, the majority of the 275 sentences (45.10%) belongs to p_+, which is higher than those that belong to p_-. We see this result as validating the effectiveness of combining the deviation statistic with the frequency statistic to identify function words even in the environment when the input word alignment is noisy.

6.4 Summary

In this chapter, we proposed a `deviate-frequent` heuristic to better approximate the `FWID` component. Although the simple `most-frequent` heuristic works well, it misses some important function words that would otherwise recommend important non-monotone reorderings. The inability to identify good heads activates either the statistics of the universal token or the content words that prefer monotone reordering. Error analyses in Chapter 5 revealed that one important type of mistakes made by the basic F W S model are indeed due to the model's overly strong bias toward monotone reordering. This motivates us to look at the orientation statistics.

We have approximated the definition of a function word as a word that appears with high frequency and suggests non-monotone reordering. To incorporate

such a hypothesis, we introduce the deviation statistic, which measures how different the orientation statistics of a head word are from those of the universal token. To get the final list of head words, we combine the frequency and the deviation statistics in a linear fashion.

Our experimental results show that our new approximation of the `FWID` component can improve the reordering performance both in the perfect lexical choice scenario and full translation task scenario. The improvement correlates with the number of genuine function words used by the model, reinforcing our hypothesis that choosing function words as heads is suitable for the reordering task.

Chapter 7

Argument Selection

This chapter concerns with improving the `ARGSEL` component, whose role in the F W S approach is to select the appropriate arguments to the heads, among all other possible sets. The basic F W S model approximates this component by employing the `immediate-neighbor` heuristic, which restricts a head's arguments only to those immediately adjacent to the head, and sets the probability of selecting each set to be equal. However, our error analysis suggests that this heuristic is suboptimal. Unfortunately, one of the basic F W S model's systematic errors concerns with the model's failure to accommodate arguments that are positioned beyond the head's immediate neighbor. This error analysis motivates us to improve the approximation to the `ARGSEL` component, allowing the head to take a more flexible set of arguments, and also to develop a statistical model to give bias toward certain set of arguments.

In retrospect, allowing a more flexible set of arguments to a head can be seen as addressing the undergeneration problem in the existing FSB model that is due to the non-adjacent nonterminal constraint, since it is equal to allowing the creation of rules with adjacent nonterminals.

7.1 Motivation

Here, we revisit our motivation to replace the `immediate-neighbor` heuristic, complementing the error analysis in Section 5.5.2. Although restricting arguments only to the head's immediate neighbor is desirable for its simplicity, here we argue that accommodating a more flexible set of arguments is important for two inter-related reasons.

First of all, the immediate neighbor restriction makes the basic F W S model asymmetric: some movements can be modeled only in one but not both sides of language. More specifically, the basic F W S model captures the movement of a function word's immediate neighbors in the source language, relocating them to the target language side as either immediate or non-immediate neighbors. However, when the translation direction is changed (i.e., swapping source and target languages), the basic F W S model will not be able to model those arguments that moved to non-immediate positions, as it is forbidden by the `immediate-neighbor` heuristic.

Secondly, there are genuine cases in language where function words must influence non-immediate neighbors. Fig. 7.1 illustrates one such case where the immediate neighbor restriction is problematic. This example represents the verbal phrase (VP) construction, which is one of the most prominent syntactic differences between Chinese and English. In particular, Fig. 7.1 illustrates a VP construction which is made up by joining a prepositional phrase (PP) and a simple VP (the one at the lowest level). When translated to English, the simple VP ends up positioned before the PP, indicating the shift from a pre-verbal construction in Chinese to a post-verbal one in English. The only function word involves in this construction is the preposition 为 (for), which – even after extending its influence to its left and right neighbors – cannot properly reorder the simple VP. The simple VP can only be reordered if the function word is allowed to take the second neighbor to its right

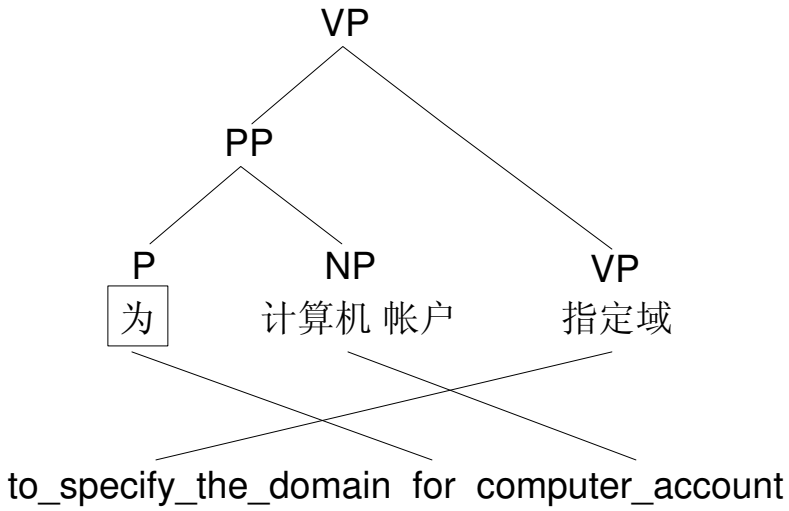

Figure 7.1: An example of the VP construction where it is vital to model non-immediate arguments. The function word involved in each example is highlighted as the Chinese character in the box. Without allowing the function word 为(for) to take non-immediate arguments, the movement of VP (为(for)'s second neighbor to its right) cannot be modeled.

as its argument.

These two reasons strengthen our motivation to replace the `immediate-neighbor` heuristic to account for a more flexible set of arguments, bringing parity in the reordering model in both source and target languages and handling real cases in language.

However, accommodating a flexible set of arguments is simple but computationally challenging as the model now has to consider more sets of arguments. Formally, letting a function word to influence its non-immediate neighbors is as easy as allowing the values of L and R of the head-driven SCFG rule to extend to more than one. Unfortunately, this effort would aggravate to the overgeneration problem, since the F W S model now needs to compute a grammar with much larger number of rules. Furthermore, it may also result in an increase in the spurious ambiguity level since it introduces rules with adjacent nonterminal, known as the main source of the spurious ambiguity (Chiang, 2005). Clearly, a new statistical model is necessary to provide the much-needed bias to certain set of arguments. In this thesis, we propose a statistical model, called *the argument selection model*, to curb both the overgeneration and the spurious ambiguity problems. In particular, this argument selection model would only encourage the grammar to choose rules with adjacent nonterminals only if these rules represent judicious uses of adjacent nonterminals. Here, we design the argument selection model to focus only on arguments that benefit the phrase reordering task.

7.2 Argument Selection Model

We design the argument selection model to model the expansion of a rule as head-outward process modeling, similar to Collins parsing model (Collins, 2003), where the head is considered to be generated first followed by the head's arguments, starting from the ones closest to the head. This process fits nicely into our decoding

implementation (see Appendix A) which requires the binarization of rules of rank more than two as such arguments are attached to the head one at a time. In particular, we refine the argument selection model into the following steps:

1. Determine T_{arg} – the total numbers of arguments for a particular head f/e – according to $P(T_{arg}|f/e)$. This probability is approximated by the *number of arguments model*: $noa(T_{arg})$, which captures the preference of (any) head to generate T_{arg} number of arguments.
2. Initialize l and r to 0, where the former is the counter for the number of left arguments generated so far while the latter is the number of right arguments generated so far.
3. If $l + r$ equal to T_{arg}, go to step 6.
4. Generate an argument X, either to the left or to the right. Update l and r accordingly afterwards while keeping the previous values l' and r'. Score the generation according to *grow* model: $grow(X, l, r|l', r', f/e)$.
5. Go to step 3.
6. Generate STOP symbols at $l + 1$ and at $r + 1$ with a score computed by the following *stop* model: $stop(l + 1|f/e)$ and $stop(r + 1|f/e)$.

As shown, the proposed argument selection model consists of three models: the number of arguments (*noa*), the grow (*grow*) and the stop (*stop*) models. The number of arguments model, as the name suggests, specifies the preference of assigning a certain number of arguments to a head. Meanwhile, the grow model specifies the preference of assigning an argument at a specific location, while the stop model captures the preference of *not* generating arguments further from a specific location onwards.

For the F W S approach, since e represents a function word, the maximum number of arguments T_{arg} should be restricted to a reasonable bound of arguments for function words. Here, we limit T_{arg} to $[0, 1, 2]$ as we observe that both in

the literature of some languages (Howard, 2002; Chino, 2001) and empirically, the influence of function words is limited up to two neighbors. Limiting T_{arg} to 2 also allows the reordering of the arguments to be trivial, i.e. in the space defined by the Inversion Transduction Grammar (ITG).

Thus, in addition to Rules 5.1-5.4 defined for the basic F W S model, the F W S model with the argument selection model use the following two rules:

$$X(h_{-2}, h_{-1}, h_Y) \rightarrow \langle X_{-2}(h_{-2})\ X_{-1}(h_{-1})\ Y(h_Y), \alpha, \sim \rangle \quad (7.1)$$

$$X(h_Y, h_{+1}, h_{+2}) \rightarrow \langle Y(h_Y)\ X_{+1}(h_{+1})\ X_{+2}(h_{+2}), \alpha, \sim \rangle \quad (7.2)$$

As an illustration, the argument selection model's score for Rule 7.1 is:

$$argsel(X(h_{-2}, h_{-1}, h_Y)) = noa(T_{arg} = 2).grow(X_{-1}, l = 1, r = 0 | l = 0, r = 0, h_Y).$$

$$grow(X_{-1}, l = 2, r = 0 | l = 1, r = 0, h_Y).stop(l{+}1 = 3 | h_Y).stop(r{+}1 = 1 | h_Y) \quad (7.3)$$

7.3 Parameter Estimation

The parameter estimation of the argument selection model involves estimating the parameters of its three components: the number of arguments (*noa*), the grow (*grow*) and the stop (*stop*) models. These models can be estimated easily if the information about the arguments of the heads are available in the training data. Unfortunately, this information is not available in the training data, thus an estimation is needed. Here, we consider all neighbors as possible arguments of a head and use the neighbors' orientation statistics as the soft count, indicating the likelihood of that neighbor to be considered the head's argument – the higher the soft count the more likely the argument is to be considered as an argument.

In using the orientation statistics, this heuristic reflects our bias towards favoring those neighbors that have great importance to the reordering task, i.e.

they move when translated just like the simpler VP in Fig. 7.1 which has to be reordered to the beginning of the phrase.

To calculate the soft counts, we put the orientation statistics of an argument in vector form: O=[$C(f/e, o = MA)$, $C(f/e, o = RA)$, $C(f/e, o = MG)$, $C(f/e, o = RG)$] and assume that there exists a contribution vector: $w = [w_{MA}$, w_{RA}, w_{MG}, $w_{RG}]$, which would reflect the model's bias towards certain orientation values. Then, we calculate the soft counts simply by performing a dot product between O and w.

The first and the second neighbors have different trivial reorderings, i.e. MA for the first neighbor and MG for the second neighbor. Thus, we use a separate contribution vector for each neighbor: w_1 and w_2, respectively. In this way, we calculate the soft counts for every function word's neighbor: D_{-2},D_{-1},D_{+1},D_{+2}, from which the model parameters can be directly estimated according to the following formulas:

$$grow(X, l, r|l', r', f/e) \approx grow(\tau|f/e) \approx \frac{D_\tau}{\sum_{\forall\tau} D_\tau}, \tau \in \{-2, -1, +1, +2\} \quad (7.4)$$

$$stop(\upsilon|f/e) \approx \begin{cases} (D_{-2} + D_{-1})/Z, \upsilon \in \{-3, +1\} \\ (D_{-1} + D_{+1})/Z, \upsilon \in \{-2, +2\} \\ (D_{+1} + D_{+2})/Z, \upsilon \in \{-1, +3\} \end{cases} \quad (7.5)$$

where Z=$2 * (D_{-1}+D_{+1}+\sum_{\forall\tau} D_\tau)$ is the stop model's normalization factor and τ is the position of the currently generated argument. Here, υ extends to -3 and $+3$ to account for the generation of the STOP symbol at $n+1$ and $m+1$. As stated, the estimation of the *grow* model is proportional to each neighbor's soft count, while the likelihood of generating a STOP symbol is proportional to the soft counts of those arguments that have been generated thus far, plus the soft counts of those potential arguments that can be generated further.

7.3.1 Parameter Estimation for Meta Parameters

The parameter estimation for the argument selection models can be performed in a relatively easy way, since it uses the statistics that are already available. The extra effort here is the estimation of the following meta parameters: the number of argument model $noa(T_{arg})$ and the contribution vectors w_1 and w_2. There are many methods to assign the values of these parameters and here, we explore two of them.

The first method is via intuition. For example, we can set the second element of w_1 to a very high value to give a preference toward selecting the first neighbor that tends to end up at reverse adjacent (RA) orientation. This method is possible because the parameters size is relatively small and the role of each element is relatively well-understood. We prepare the following sets of values: $noa = [0.01, 0.14, 0.85]$, $w_1 = [0.25, 2.00, 0.15, 0.15]$ and $w_2 = [0.3, 1.0, 0.1, 0.25]$, which reflect our bias toward assigning as many arguments as possible and assigning the second neighbor argument if it tends to move to a non-trivial orientation.

The second method explored is via automatic training, where we treat these parameters as latent variables whose values will be estimated automatically from the statistics of the development set. Eventually, such a procedure will find a set of parameters that optimizes a certain training criterion. A standard method to approach such a latent variable problem is to use Expectation Maximization (EM) (Dempster, Laird, and Rubin, 1977). However, here we opt to use a much simpler method since as shown in the experiments, the meta parameters produced by this ad-hoc method performs on par with the meta parameters produced by human intuition. Nevertheless, we intend to explore a more principled method to estimate these parameters in the future.

In particular, we devise a simple training criterion that indicates the parameter's contribution to the task of selecting arguments – which is the intended

Annotation	#fw	#instances
-	54	90
-1	77	310
+1	421	1,546
-1 +1	140	1,321
+1 +2	27	40
M	21	57
M +1	2	2
Total	209	3,371

Table 7.1: Statistics of the annotation extracted from the 500 sentence pairs which are part of the development set. The first column indicates the annotation, while the second and third column indicate the number of distinct function words and the number of instances that received the annotation specified in the first column, respectively.

use of the argument selection model. Our automated approach needs access to gold standard function word arguments to extrapolate the parameters. For this purpose, we asked an expert Chinese linguist to annotate the genuine arguments of function words in the first half of the development set (500 sentences). Here, we used the gold standard function words, described in Chapter 4. The linguist then annotated each function word with its arguments, by first identifying it and then labeling it with one of the following position labels (...,−2,−1,+1,+2,...). The data collected amounts to a total of 209 function words (inclusive of split function words; e.g. the function word 从...上, translated to "from" in English).

Table 7.1 shows the statistics of the annotation while Table 7.2 shows an excerpt of the annotation supplied by the linguist. Note that the label M refers to the argument that is in the middle of a split function word, as exemplified in Table 7.2. In Table 7.1, the linguist annotated 56.75% of all function words as taking a single argument (either −1 or +1), 40.43% as taking two arguments, and a small percentage (2.81%) to either having zero or three arguments. The table also shows that the majority of function words take their immediate neighbors as

their arguments and that only a small minority take the second neighbor.

Chinese:	而/0 在/1 使用/2 这些/3 辅助/4 功能/5 时/6 ， /7 无需/8 购买/9 任何/10 特殊/11 的/12 设备/13 。/14
English:	you/0 do/1 not/2 need_to/3 purchase/4 any/5 special/6 equipment/7 to/8 use/9 these/10 features/11 ./12
Alignment:	(2-9); (3-10); (5-11); (8-2); (8-3); (9-4); (10-5); (11-6); (13-7); (14-12);
Annotation:	[而/0] : +1 [在/1] : +1 [的/12] : -1 +1 [在...时/1,6] : M

Table 7.2: A sample of sentence pair annotated with function words and their arguments. Note that the English and Chinese words are indexed and their correspondences are available in the third line. The last function word represents a split function word. -1 refers to the first neighbor to the left, +1 the first neighbor to the right, while M the argument in the middle of a split function word.

We then treat these annotation as a list of the following tuples: $(f/e, a)$ where $a \in \{\ \emptyset,\ -1,\ +1,\ -1+1,\ -2-1,\ +1+2,\ M,\ M+1\ \}$. The estimation of the *noa* parameters is obtained from the tuple count $C(f/e, a)$ as follow:

$$noa(a_{\#arg}) = \begin{cases} \underset{\forall a_0}{C}(f/e, a_0)/C(f/e, \cdot) = 0.0281 & , a_0 \in \{\emptyset\} \\ \underset{\forall a_1}{C}(f/e, a_1)/C(f/e, \cdot) = 0.5674 & , a_1 \in \{-1, +1, M\} \\ \underset{\forall a_2}{C}(f/e, a_2)/C(f/e, \cdot) = 0.4043 & , a_2 \in \{-2-1, -1+1, +1+2, M+1\} \end{cases} \tag{7.6}$$

To estimate w_1 and w_2, we devise the following objective function:

$$w_1', w_2' = \underset{w_1, w_2}{\operatorname{argmax}} \sum_{\forall f/e, a} \delta(a; m, n = \underset{l'', r''}{\operatorname{argmax}}\ argsel(l'', r''|noa, w_1, w_2, f/e)) \tag{7.7}$$

In Eq. 7.7, l'', r'' is the annotation assigned by the argument selection for f/e given certain contribution vectors (w_1 and w_2) and previously estimated *noa* models;

while δ is a Kronecker delta function which outputs 1 if l'', r'' matches the human annotation a, otherwise 0.

To find the optimum w_1 and w_2, we devise a simple grid search algorithm that takes a single parameter and greedily optimizes it, repeating as necessary:

1. Define an initial value. Random values are used in our experiments.

2. Define a discrete space for each parameter by setting the minimum and maximum value together with their resolution. For the reported experiment, we defined the minimum value to be 0.001, the maximum value to be 4.0, while the resolution to be 0.001 for each parameter.

3. Define flags to keep a record of all unmodified parameters. The flags are all initialized as unmodified.

4. For every remaining unmodified parameter p, explore the parameter space defined for p while fixing the other parameters at the value stored in the current state of w_1 and w_2. At any point, the algorithm evaluates the objective function and records the point that gives the maximum value.

5. Pick the one parameter that gives the best improvement, update the flag of that parameter to be modified and set the corresponding value in either w_1 or w_2 with the best point.

6. If there are still unmodified parameters, return to step 3; otherwise, terminate. Finally, the algorithm outputs w_1 and w_2 which are the parameters that give the optimal value with respect to the objective function.

In short, the algorithm updates the parameter that gives the best improvement, one parameter at a time until all the parameters are visited.

The following values are the results: $w_1 = [2.993, 2.521, 0.202, 0.15]$ and $w_2 = [0.144, 1.315, 0.25, 0.249]$. When we manually checked the model output,

we observed that most errors concern with cases of selecting two arguments. One reason is evident in the value of $noa(T_{arg} = 1)$, which is higher than $noa(T_{arg} = 2)$. Thus, this set of parameters gives a strong bias toward selecting only one argument. Unfortunately, such a bias greatly penalizes the F W S approach since fewer arguments are then influenced by function words.

Thus, while being able to identify linguistically-motivated arguments is desirable, we opted to alter the definition of arguments to be reordering-centric. Specifically, we performed the following transformations to the annotation:

1. Change all annotations of split function words to take zero-arguments.

2. Duplicate all instances of 1-argument function words (-1 and $+1$) and annotate the copies as 2-arguments function words ($-1 + 1$).

By applying these transformations, we hope to make the heads to take as many arguments as possible and to select the linguistically-motivated arguments as well.

When we ran the same procedure over the transformed set, the following values were obtained: $noa = [0.018, 0.291, 0.691]$, $w_1 = [0.249, 2.057, 0.15, 0.02]$ and $w_2 = [0.206, 3.375, 0.001, 0.249]$. As shown, these parameters are more in line with the manually-set parameters, indicating the same bias toward assigning arguments that exhibit non-monotonic reorderings. For instance, both in w_1 and w_2, the weight for reverse orientation (the second element) is significantly larger than the weight for monotone orientation (the first and the third elements). In w_2, the weight of the trivial orientation – which corresponds to monotone gap (the third element) – is relatively small. We can interpret this value as the argument selection model that would avoid selecting the second neighbor as an argument unless it exhibits a non-trivial reordering. This also means that the F W S model would avoid applying rules with adjacent nonterminals which target language order is exactly the same as the source language order.

7.4 Experiments

In this section, we evaluate our proposed approximation to the ARGSEL component with the following goals: 1) to study whether our argument selection model is able to fix the basic F W S model's second error that concerns with the failure of correctly reordering arguments beyond the head's immediate neighbor; 2) to evaluate whether our proposal improves (or decreases) the reordering quality; and 3) to verify whether our proposal gives the similar improvement (or drop) in the full translation task scenario. Note that the experiments in this chapter are independent of the experiments in Chapter 6.

We pursue the first and the second goals in the perfect lexical choice scenario in Section 7.4.1, and the third one in the full translation task scenario in Section 7.4.2. Specific to the pursuit of the second goal, we used the `unhandled-arg` metric described in Section 5.5.2 which measures the number of `pORI-acc` mistakes that are due arguments beyond the immediate neighbor. For `unhandled-arg` metric, lower is better; while for the `pORI-acc`, higher is better.

7.4.1 Perfect Lexical Choice

Table 7.3 shows the results of this set of experiments on different number of lexical items N. We couple the results with Table 7.4 which shows the statistics of the arguments assigned to the heads. Note that Table 7.4 only reports the statistics of the systems where N=128.

In the tables, the *ori* row represents the baseline, taken from Chapter 5, where only the immediate neighbors of the function words are considered; the *ori+noargsel* row represents the F W S model which accommodates more flexible arguments but employs no argument selection mechanism. Meanwhile, the *ori+argsel_manu* and *ori+argsel_auto* represent the F W S models that accommo-

	N=	1	4	16	64	128	256	1,024
unhandled arg	*ori*	2,251	2,210	2,167	2,153	2,080	2,042	2,044
	ori+noargsel	2,282	2,283	2,242	2,146	2,067	2,072	2,065
	ori+argsel_manu	2,230	2,181	2,157	2,067	2,014	**1,965**	2,023
	ori+argsel_auto	2,253	2,184	2,152	2,044	*1,975*	2,018	2,018
BLEU	*ori*	77.68	77.78	78.44	79.00	79.58	80.11	80.07
	ori+noargsel	78.08	78.20	78.92	79.74	79.87	79.99	79.85
	ori+argsel_manu	77.94	78.33	79.08	79.83	80.17	80.33	80.13
	ori+argsel_auto	77.89	78.32	79.04	79.91	*80.35*	**80.46**	80.20

Table 7.3: The number of `pORI-acc` errors that are classified as `unhandled-arg` of the perfect lexical choice for different argument selection mechanism along with their BLEU scores. The best score is in **bold**.

date more flexible arguments which meta parameters are estimated from manual (intuition) and automatic methods, respectively.

As shown in the baseline *ori* row, the number of `unhandled-arg` errors decreases as N increases until a certain point where it reaches a plateau, suggesting that increasing the number of lexical items modeled further cannot reduce this type of error. This result suggests that modeling non-immediate arguments is beneficial only for some small cases (related to function words) but not so for the majority of cases. Additionally, data sparseness may interfere the performance of the system as the orientation statistics for low frequency words may not be reliable enough to capture the word's true orientation statistics.

When we analyzed the impact of allowing a more flexible set of arguments without employing the argument selection model, we observe only limited error reduction, shown in the *ori+noargsel* row. In Table 7.4, the *ori+noargsel* assigns a significant amount of cases to the second neighbor arguments (the last two columns), more than other systems. We suspect that such an aggressive assignment contributes to the error increase in other cases.

Higher error reduction can be obtained by employing the argument selec-

System	$\emptyset$	X_{+1}	X_{-1}	$X_{-1}X_{+1}$	$X_{+1}X_{+2}$	$X_{-2}X_{-1}$
ori	2,878	2,622	1,855	8554	0	0
ori+noargsel	3,385	1,061	702	4,216	2,628	3,917
ori+argsel_manu	4,475	1,636	1,679	5,540	1,682	897
ori+argsel_auto	1,488	4,154	5,232	4,069	672	294

Table 7.4: Statistics of the arguments assigned by different argument selection mechanism in the perfect lexical choice scenario. The number of heads used is $N = 128$.

tion mechanism. Moreover, the most effective error reduction can be obtained by employing the argument selection mechanism where the meta parameters are automatically estimated at $N = 128$ as indicated by *ori+argsel_auto*. The same trend is also apparent in terms of BLEU score. Note that the performance difference between the argument selection model with automatic and with manual estimation of the meta parameters is relatively insignificant, although the latter seems to be consistently better. We suspect that this is because the latter (*ori+argsel_auto*) assigned second arguments more conservatively than the former (*ori+argsel_manu*) as shown in Table 7.4. We see this result as validating our hypothesis that we can improve the F W S model by allowing function words to take more flexible arguments.

7.4.2 Full Translation Task

Here, we investigate whether our proposed argument selection model is robust enough when the input word alignment is noisy. As the baseline, we employed the basic F W S model with the $N = 128$ most frequent words as the heads, as this setup gives the most efficient performance gain in the perfect lexical choice scenario. Table 7.5 reports the performance of the proposal in this scenario.

As shown, the same performance trend as in the perfect lexical choice sce-

System	BLEU
ori	24.92
ori+noargsel	24.98
ori+argsel_manu	25.44
ori+argsel_auto	25.59

Table 7.5: BLEU scores for the full translation task where sets of flexible arguments are used.

nario, is observed. The best performing system *ori+argsel_auto* is able to produce a modest improvement over the baseline *ori* system. To better understand the result, we analyzed the output by looking at the three sets produced as the intermediate results of the statistical significance test. Specifically, we analyzed the intermediate results of the statistical significance test when we compare *ori+argsel_auto* and *ori* to look at the contribution of allowing the second neighbor argument.

Set	Sign test	2_{nd} neighbor	
	Count	Count	%
$\mathbf{p}_+$	586	419	32.06
p_0	939	534	40.86
p_-	475	354	27.08
Total	2,000	1,307	100

Table 7.6: The comparison between *ori+argsel_auto* and the baseline *ori*. p_+ refers to *ori+argsel_auto*>*ori*, p_- refers to *ori+argsel_auto*<*ori*, while p_0 refers to *ori+argsel_auto* = *ori*. The column labeled "2_{nd} neighbor" refers to the number of sentences in each set that uses rules with second neighbor arguments. Between p_+ and p_-, the one with more sentences is in **bold**.

We report our analysis in Table 7.6. In particular, we show the number of translations involving the second neighbor arguments. In total, rules with second neighbor arguments were involved in translating 1,307 test sentences. Out of these sentences, the majority of them appears in p_0 (40.86%) where the performance of

the systems is equal. A sizable amount of sentences (32.06%) appears in p_+ where *ori+argsel_auto* performs better than *ori*, higher than in p_- (27.08%). To a certain extent, this suggests that our proposed argument selection model correlates well with the improvement gain reported in Table 7.5. We see this as validating our claim that modeling a more flexible set of arguments is beneficial not only to the reordering task but also to the translation task.

7.5 Summary

This chapter is centered around our effort to allow a more flexible set of arguments to the head, improving the basic F W S model's approximation to the `ARGSEL` component, which rigidly specifies that the head can only take arguments that are located next to the head. In summary, we have argued for a more flexible set of arguments, i.e. allowing the function word's arguments to take the position of a non-immediate neighbor, empirically and using real examples. In retrospect, this effort is equal to addressing the undergeneration problem of the existing FSB models, since it allows the creation of rules with adjacent nonterminal.

While well-motivated, injecting flexibility in selecting the arguments unfortunately raises the spurious ambiguity concern as it is equal to introducing rules with adjacent nonterminals which are deemed as the spurious ambiguity's main source. Here, the model faces a trade-off between dealing with the undergeneration problem or keeping the level of ambiguity as low as possible. In this thesis, we prefer the latter by proposing the argument selection model that assigns arguments to a head based on how the arguments move in the training data, utilizing the idea of head-outward process from the Collins parsing model (Collins, 2003). Our idea in developing such an automatic argument selection model is to select only those arguments that are likely to move during translation process. Our experimental results show that such modeling improves translation quality.

Chapter 8

Order of Rule Application

In this chapter, we concern with improving the FWORDER component, whose responsibility is to arrange the order of rule application. The basic F W S model uses the function word's unigram probabilities to approximate this component, favoring to apply rules headed by more frequent words first. However, our error analysis in Section 5.5.3 shows that such an approximation is suboptimal, partly because resolving the order of rule application may require contextual information. In this chapter, we propose to replace the preference model with a more context-sensitive model, called *the pairwise dominance model*.

In developing the pairwise dominance model, our upcoming effort can be seen as addressing the overgeneration problem, since the pairwise dominance model gives bias toward selecting derivations that hopefully lead to the correct reordering. This model also demonstrates the strength of the head-driven SCFG in its use of the lexicalization of the nonterminals, which represents one key difference of the head-driven SCFG with the existing SCFGs.

8.1 Motivation

Fig. 8.1 illustrates a concrete example of the overgeneration problem. Suppose a grammar is defined over the example, consisting of the following three rewrite rules: (i) $X \rightarrow \langle X_{-1}$ 和 X_{+1}, X_{-1} and $X_{+1}\rangle$; (ii) $X \rightarrow \langle X_{-1}$ 是 X_{+1}, X_{-1} are $X_{+1}\rangle$; and (iii) $X \rightarrow \langle X_{-1}$ 的 X_{+1}, X_{+1} of $X_{-1}\rangle$. This grammar has all other ambiguities resolved with the exception of the application order of the rules. Note that this grammar resembles rules in a typical FSB model.

Focusing on Rules (ii) and (iii), one can see that there are two possible orders. The grammar can either apply Rule (iii) before Rule (ii), making Rule (ii) the parent of Rule (iii), or vice versa. While the former leads to the correct translation (Fig. 8.1a), the latter creates an incorrect noun phrase that constitutes the copula 是 (are) (Fig. 8.1b). To resolve such ambiguities, clearly we must incorporate the information from both rules, e.g. the head of the dominating rule and the head of the dominated one.

Getting the information about the order of rule application is not possible in the example grammar, as all the rules use a single generic nonterminal X homogenously and no information is available beyond the rule. In particular, this notation exposes no information about the children nodes to the parent nodes. Here, we exploit the fact that the head-driven SCFG propagates the information about function words from child nonterminals to its parents, which can be used to resolve the rules' order of application.

In the head-driven SCFG, the two orders of application involve two different sets of grammars. The head-driven SCFG equivalent for the above grammar that would yield the *incorrect* order of application would consist of the following rules: (iv) X(和/and) $\rightarrow \langle X_{-1}$ 和 X_{+1}, X_{-1} and $X_{+1}\rangle$; (v) X(和/and, 是/are) $\rightarrow$ $\langle X_{-1}$(和/and) 是 X_{+1}, X_{-1}(和/and) are $X_{+1}\rangle$; and (vi) X(和/and, 是/are, 的/of) $\rightarrow$ $\langle X_{-1}$(和/and,是/are) 的 X_{+1}, X_{+1} of X_{-1}(和/and,是/are)$\rangle$.

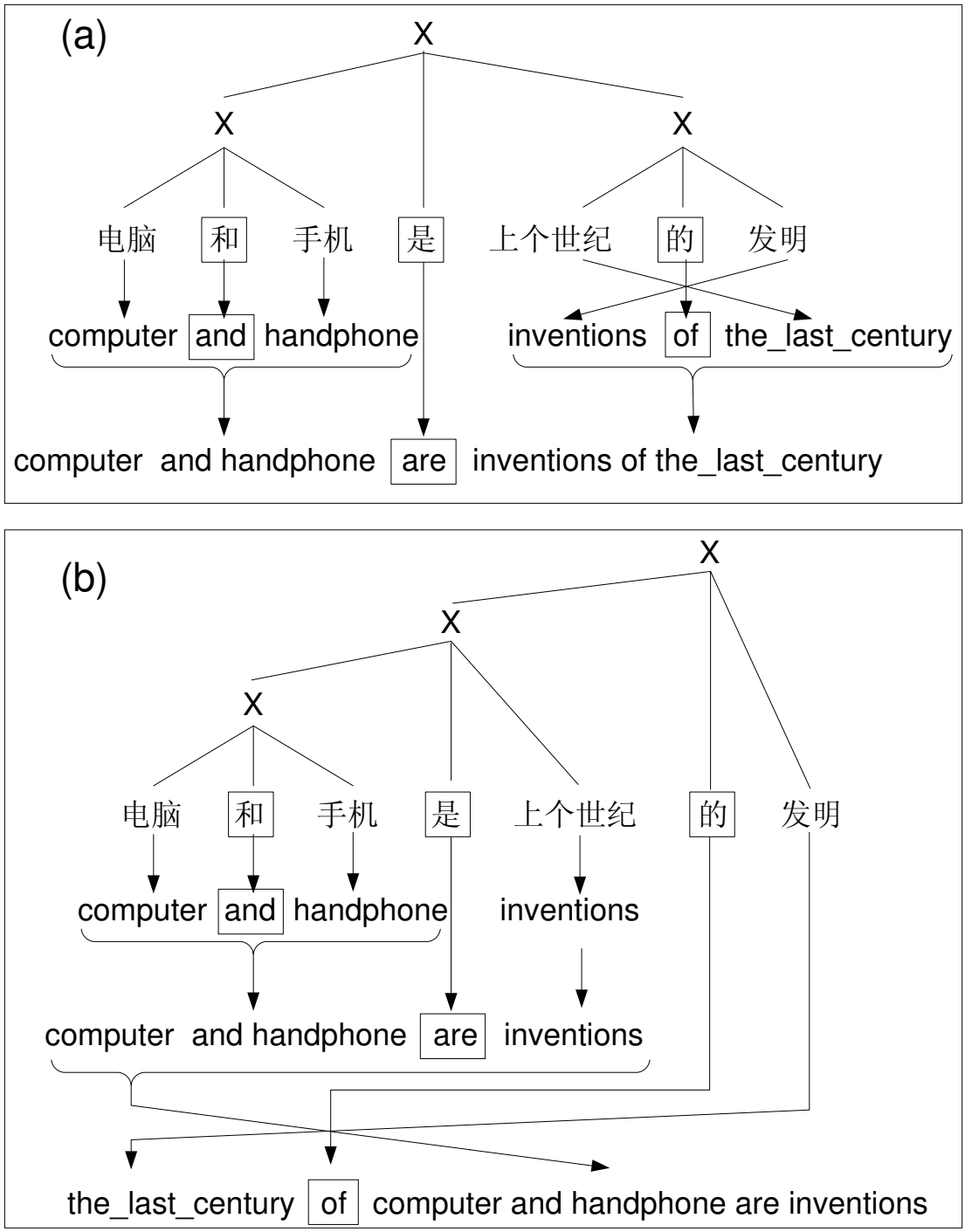

Figure 8.1: Instances of applying SCFG rules in **a**) the correct order and **b**) the incorrect order.

Meanwhile, the head-driven SCFG equivalent for the above grammar that would yield the *correct* order of application would consist of the following rules: (vii) X(和/and) $\rightarrow \langle X_{-1}$ 和 X_{+1}, X_{-1} and $X_{+1}\rangle$; (viii) X(和/and,是/are,的/of) $\rightarrow \langle X_{-1}$(和/and) 是 X_{+1}(的/of), X_{-1}(和/and) are X_{+1}(的/of)$\rangle$; and (ix) X(的/of) $\rightarrow \langle X_{-1}$ 的 X_{+1}, X_{+1} of $X_{-1}\rangle$.

Clearly, the two sets of rules are different. Comparing Rules vi and ix which model the swapping of phrases around the word 的(of), we notice that the function word in Rule vi incorrectly takes a left argument that spans the other two function words: 和 (and) and 是 (are) indicated in the lexicalization of X_{-1}; while in constrast, the function word in Rule ix takes a left argument whose span does not include the two other function words. The upcoming pairwise dominance basically exploits this information, i.e. that in this case, the rule headed by the function word 的(of) should be applied much latter (thus takes no argument that spans 和 (and) and 是 (are)).

8.2 Pairwise Dominance Model

Exploiting the lexicalization in the head-driven SCFG, we propose the *pairwise dominance model (dom)* as an approximation of the `FWORDER` component. The goal of this model is to specify the correct order of application given two competing rules, i.e. which rule should become the parent of another.

We develop this pairwise dominance model by first developing the `pORD` function. In particular, we design this function to output four dominance values `pORD`(h', h'') that takes the two rules' set of heads (h' and h'' where h' precedes h'' in the source text) as inputs and produces one of the following four dominance values: {left, right, either, neither} as output. These four dominance values basically specify which of the two rules should be applied first, thus appears higher than the other in the hierarchical structure. For the left value, it is the rule headed by h';

for the right value, it is the rule headed by h''; for the either value, it is either h' or h''; while for the neither value, it is none of the two.

The left value is exemplified in Fig. 8.1a, where the rule headed by the copula 是 (are) must appear above the rule headed by the particle 的 (of). Meanwhile, the `either` is illustrated in Fig. 8.1a, where applying either Rule (i) or (ii) first does not change the final word order. The neither value refers to cases where none of the two rules should have dominance, which models cases where the two function words do not share a common parent.

Once the `pORD` function is defined, we can directly develop the pairwise dominance model. Specifically, the pairwise dominance score for a rule is equal to the sum of the `pORD` probabilities between the rule's head with each of its arguments. Thus formally, the pairwise dominance model takes the following form:

$$dom(X(h_{-L}, ..., h_{-1}, h_Y, h_{+1}, ..., h_{+R})) \approx$$
$$\prod_{l=1}^{L} P(\texttt{pORD}(h_{-l}, h_Y)|h_{-l}, h_Y). \prod_{r=1}^{R} P(\texttt{pORD}(h_Y, h_{+r})|h_Y, h_{+r}) \tag{8.1}$$

Note that the appearance of the lexical heads matters.

8.3 Parameter Estimation

Like all other models in this thesis, estimating the parameters of the pairwise dominance model involves approximating information not directly seen in the training data. Ideally, learning this model's parameters would require information about the hierarchical structure, from which the dominance relation can be counted, as such the probability of a dominance value can be easily estimated. However, such a hierarchical structure is unavailable in FSB models.

We approximate the dominance relationship by making several simplifying assumptions. First of all, we approximate the formula in Eq.8.1 so that it only

compute the dominance relationship between two bordering function words that come from different rules. More concretely, we approximate the formula in Eq.8.1 into:

$$dom(X(h_{-L}, ..., h_{-1}, h_Y, h_{+1}, ..., h_{+R})) \approx$$

$$\prod_{l=1}^{L-1} P(\texttt{pORD}(last(h_{-(l+1)}), first(h_{-l}))|last(h_{-(l+1)}), first(h_{-l})).$$
$$P(\texttt{pORD}(last(h_{-1}), h_Y)|last(h_{-i}), h_Y)).$$
$$P(\texttt{pORD}(h_Y, first(h_{+1}))|h_Y, first(h_{+1}))).$$
$$\prod_{r=1}^{R-1} P(\texttt{pORD}(last(h_{+r}), first(h_{+(r+1)}))|last(h_{+r}), first(h_{+(r+1)})) \quad (8.2)$$

where $first$ and $last$ are the functions that give the first and the last element of h respectively. Because of this assumption, the dominance values between two heads may not necessarily be an immediate parent-children relationship but ancestral. With this approximation, each factor is calculated whenever an argument is attached to the head.

Secondly, we assume that we can recover the dominance relationship between two function words using alignment information, which can be observed in the training data. The idea is that each different dominance relationship correspond to different phrase alignment configuration. Specifically, we return to the consistent alignment heuristic, previously used for the orientation model training, as a way to identify the different phrase alignment configuration caused by the different dominance relationship.

More concretely, we first define *Maximal Consistent Head Alignments* (hereafter MCHA) which is the consistent alignment that starts from or ends with the head in the source language. The maximal sense is required to ensure the uniqueness of the phrase alignment of a head. Note that there are two MCHAs for each function word: one that ends with the function word and the other that starts from the function word.

Given two function word heads f' and f'' in the source text, the pORD value is defined by examining the MCHA of the two heads as follows

$$\texttt{pORD}(f', f'') = \begin{cases} \textsf{left}, & f' \notin \text{MCHA}(f'') \wedge f'' \in \text{MCHA}(f') \\ \textsf{right}, & f' \in \text{MCHA}(f'') \wedge f'' \notin \text{MCHA}(f') \\ \textsf{either}, & f' \in \text{MCHA}(f'') \wedge f'' \in \text{MCHA}(f') \\ \textsf{neither}, & f' \notin \text{MCHA}(f'') \wedge f'' \notin \text{MCHA}(f') \end{cases} \tag{8.3}$$

Fig. 8.3a illustrates the left value where the intersection of both MCHAs contains only the second head (f''). Meanwhile, Fig. 8.3b illustrates the either value where the intersection contains both heads. Similarly, right is represented by an intersection that contains only the first head (f'), while neither is represented by an empty intersection.

Once the counts $C(\texttt{pORD}(f', f''))$ are computed, the pairwise dominance model can be estimated according to a maximum likelihood principle as follows:

$$dom(\texttt{pORD}(f', f'') = \rho | f', f'') \approx \frac{C(\texttt{pORD}(f', f'') = \rho)}{\sum_{\forall \rho'} C(\texttt{pORD}(f', f'') = \rho')} \tag{8.4}$$

where $\rho \in \{\textsf{left},\textsf{right},\textsf{either},\textsf{neither}\}$.

8.4 Decoding

This section concerns with the question of how this model actually works during decoding time? The answer to this question is simple: the pairwise dominance model behaves like an n-gram language model since both are stateful features. Stateful features refer to those features that require extra information beyond the span under consider in their computation. In particular, the pairwise dominance model can be seen as a bigram (n=2) model, except compared to the bigram language

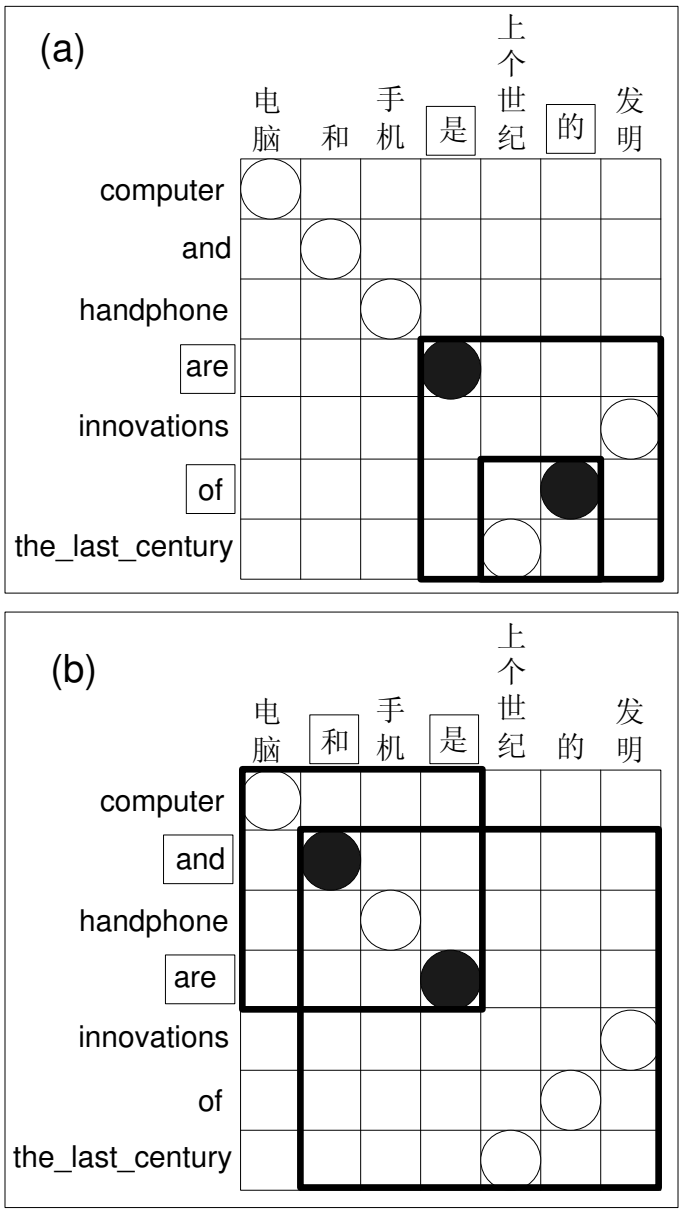

Figure 8.2: Illustrations for: a) the left value, where the rule headed by the copula 是(are) must be applied at the level higher than the rule headed by the particle 的(of); b) the either value, where the rules headed by either head tokens (和(and) and 是(are)) can applied in any order. The MCHAs of the two head tokens are in thick outlined boxes while the two head tokens' alignment points are indicated as solid circles. The intersections of the two MCHAs are in the gray box.

model, it requires extra information as context. The two differ in the terms of the information they use as context.

In order to calculate a language model score of a word, the decoder needs to the previous $n-1$ words as context. Meanwhile, in order to calculate a pairwise dominance model score of a function word Y'', the decoder needs the previous function word Y' as context as well as the relevant alignment information. Specifically, during decoding time, the decoder must pass the following information: the first and the last function words given a particular span along with the relevant alignment information – the first function word is the function word which dominance score has yet to be computed, while the last function word is the context for computing the dominance score of the next function word. Thus, employing the pairwise dominance model requires no significant changes to the design of the F W S decoder, i.e. the same CKY decoder as described in Appendix A is used.

8.5 Position-sensitive Pairwise Dominance Model

We can extend the pairwise dominance model to incorporate more diverse contextual information. Here, we explore one possible extension which looks at the position of the head in the source sentence. The motivation is that function words may have different roles in syntax and at a certain position, they tend to have only a specific role. Before discussing the statistics that motivate this extension, we first develop the model.

To take the positional information into account, we develop the position-sensitive pairwise dominance model (*domp*). Recall that the head-driven SCFG keeps the ordering of the heads in the source language when it propagates the head information to the higher level of structure. As such, the root of the hierarchical structure contains the following information $X(h_{-L}, ..., h_{-1}, h_Y, h_{+1}, ..., h_{+R})$, where each h is a list of function words or can be an empty list. Expanding the

list, the information inside the bracket is equal to a sequence of function words $(f@0, f@1, f@2, ..., f@F)$ where the order of appearance in the source text is maintained by the index following the "@" symbol. Note that this index is the function word's index with respect to the function word list and not to be confused with the word's position in the source sentence.

Here, we are interested in two groups of function words: the ones near the beginning of the sentence and the ones near the end of the sentence. As we will show shortly, these groups of function words often exhibit some interesting statistics, which we will discuss soon. In particular, we observe that the dominance values of these function word pair in the middle of the sentence are quite similar to the original model, thus separate model is not needed.

Table. 8.1 shows the dominance statistics of the function word 的 (of) with and without positional information. As shown, the prevailing statistic of the function word in the original model is the either value. However, when we incorporate the positional information, we observe different more fine-grained statistics. For instance, if we look at the statistics when the word acts as the preceding head and appears near the end of the sentence (thus 的@$F-1$), then its prevailing dominant value is right, suggesting that the function word should appear at the level lower than its succeeding function word. This takes into account that the last word in almost all sentences in the corpus is a period (.) – the third most frequent word in the corpus –, which is unlikely to move. Similarly, if we look at the statistics of the same function word when it acts as the succeeding word and appears near the start of the sentence (thus 的@$F-1$), then its prevailing statistic is right, suggesting that the word should appear at the level higher than the preceding word. This often corresponds to the the corpus-specific tendency of creating a long noun phrase at the end of the sentence, as we observe in the HIT corpus.

The formula as well as the parameter estimation of the position-sensitive

f'	f''	neither	right	left	either
的	f''	4.74	25.87	15.98	**53.41**
的@0	f''@1	2.41	10.84	18.07	**68.67**
的@1	f''@2	4.60	23.40	15.80	**56.20**
的@$F-1$	f''@F	0.25	**68.70**	1.27	29.78
的@$F-2$	f''@$F-1$	4.39	13.52	19.87	**62.22**
Y'	的	3.62	29.58	13.93	**52.86**
f'@0	的@1	1.54	19.12	25.05	**54.29**
f'@1	的@2	3.55	26.35	17.06	**53.04**
f'@$F-1$	的@F	5.48	**45.21**	5.48	43.84
f'@$F-2$	的@$F-1$	3.53	36.20	15.48	**44.80**

Table 8.1: The position-sensitive and the original pairwise dominance values for the function word 的(of). Here, the statistics are obtained by collapsing the competing function words. The position of the word is indicated by the index following "@" symbol. The most probable dominance value is in **bold**.

pairwise dominance model are exactly the original pairwise dominance model, except that we attach additional positional information to the head and that we ignore counting the statistics of those pairs that appears in the middle of the sentence.

8.6 Experiments

We set the goals for the experiments in this section as follows: 1) to study the effect of our proposed dominance model on the third error of the basic F W S model (incorrectly assign order of rule application), 2) to evaluate our proposal in terms of the reordering quality; and 3) to verify whether the effect of the proposed model resonates to the scenario where the input is trained on noisy word alignment. We divide our inquiry into two sections where we concentrate on pursuing the first two goals in Section 8.6.1 while reserving the last one to Section 8.6.2.

In pursuing the first goal, we specifically devised one metric which we subsequently refer to as `pORD-acc`. This metric measures how accurate is the model in

assigning the pORD predicate to every pair of function words in the source text. For pORD-acc metric, higher is better. In our evaluation, we used the gold standard function words described in Chapter 4 to facility fair comparison across different systems. Note that the experiments reported here are independent of those in Chapters 6 and 7.

8.6.1 Perfect Lexical Choice

Here, we compare the preference model with the proposed pairwise dominance model to study the effect of our proposed model, in the case where the word alignment is correctly given. In Table 8.2, the baseline preference model is represented by *ori+pref*, the dominance model by *ori+dom* while the position-sensitive dominance model by *ori+domp*. We also show the performance of the basic F W S model without the preference model in the *ori* row as a reference. Similar to the experiments in the previous chapters, we evaluated the system with different number of lexical items N.

Comparing the basic F W S model with and without the preference model, i.e. the *ori+pref* and *ori* rows, we can observe that the pORD-acc of the preference model drops across different N. However, when we employ the dominance model, the pORD-acc increases quite significantly. Employing the position-sensitive dominance model alone also gives an increase in accuracy but only modestly, perhaps because this model only looks at smaller set of heads in each test sentence. Employing both the position-sensitive dominance model with the position-insensitive one gives an additional increase in the accuracy. The same trend also applies to the BLEU score. We are pleased with these results since they confirm that our approximation to order of rule application resolution leads to better overall reordering quality.

	N=	1	4	16	64	128	256	1,024
pORD-acc	*ori*	75.19	75.34	75.90	76.49	76.69	77.49	77.16
	ori+pref	74.75	74.73	75.24	75.91	75.91	76.59	75.96
	ori+dom	-	75.47	77.66	77.87	78.65	78.40	77.55
	ori+domp	-	76.08	77.35	76.90	77.27	77.69	77.09
	ori+dom+domp	-	75.42	77.68	77.91	**78.72**	78.37	77.52
BLEU	*ori*	77.68	77.78	78.44	79.00	79.58	80.11	80.07
	ori+pref	77.77	78.23	78.65	79.41	79.69	80.07	80.17
	ori+dom	-	77.84	79.19	80.05	80.85	**81.26**	81.20
	ori+domp	-	77.88	78.96	79.34	79.80	80.17	80.05
	ori+dom+domp	-	77.82	79.20	80.13	80.90	81.25	81.13

Table 8.2: BLEU scores and pORD-acc of the F W S model with perfect lexical choice for different experimental setups. The best score is in **bold**.

8.6.2 Full Translation

Here we replicated the same experimental settings as in the perfect lexical choice, but with the added lexical-related ambiguities. We would like to understand whether the same favorable improvement in the perfect lexical choice scenario also applies to this scenario. As shown in Table 8.3, the same trend as in the perfect lexical scenario is reported for the full translation task.

We analyzed the intermediate results of the statistical significance test to better understand the improvement gain. In particular, we are interested in analyzing whether the dominance model is responsible for the performance gain given by *ori+dom+domp* over *ori+pref*. To do so, we recorded the sentences where the dominance values (pORD) of *ori+dom+domp* differ from those in *ori+pref*. Note that we resorted to this approximation since the true dominance values in this scenario could not be obtained.

Table 8.4 shows the statistics of the three sets together with the number of sentences in each set which pORD values differ. In total, there are 871 sentences where the two systems have different pORD values. Out of these sentences, the ma-

System	BLEU
ori	24.92
ori+pref	25.06
ori+dom	25.64
ori+domp	25.24
ori+dom+domp	25.79

Table 8.3: BLEU scores for the full translation task. *ori* represents the model taken from Chapter 5, *ori+pref* represents the baseline model, coupling the orientation model with the preference model; *ori+dom* the orientation model coupled with the dominance model; *ori+domp* the orientation model coupled with the position-sensitive dominance model; while *ori+dom+domp* the orientation model coupled with the both dominance models.

Set	Sign test	pORD-diff	
	Count	Count	%
$\mathbf{p}_+$	554	396	45.46
p_0	1,012	177	20.32
p_-	434	298	34.22
Total	2,000	871	100

Table 8.4: The comparison between *ori+dom+domp* and *ori+pref*. p_+ refers to *ori+dom+domp*>*ori+pref*, p_- refers to *ori+dom+domp* < *ori+pref*, while p_0 refers to *ori+dom+domp*=*ori+pref*. The pORD-diff column refers to the number of sentences in each set which pORD values differ.

jority of them (45.46%) belongs to p_+, which is higher than the number of sentences that belongs to p_- (34.22%). We see this result as validating our hypothesis that our approximation for resolving the order of rule application is beneficial in both the perfect lexical choice and the full translation task scenarios.

8.7 Summary

This chapter centers around our effort to improve the approximation to the FWORDER component. Our effort here is equal to addressing the overgeneration problem in the existing FSB models, by providing a bias to the model toward selecting the derivation with the most appropriate order of application. The basic F W S model introduced the preference model which uses the unigram probability of the dominating heads, but our previous analysis suggested that this model is suboptimal as it only uses limited contextual information.

In this chapter, we utilized the lexicalization feature of the head-driven SCFG, which propagates the head information to the higher level hierarchical structure. In particular, we developed a pairwise dominance model, which in a nutshell, creates a topological order of rule by looking at the phrase alignment around every pair of heads. We have shown through our experimental results that the proposed pairwise dominance model performs well, confirming our hypothesis that resolving the order of rule application is beneficial to the reordering task.

Chapter 9

The Improved F W S model

In this chapter, we develop and report the experiments of the improved F W S model. The improved F W S model replaces the basic F W S model's approximation to the `FWID`, `ARGSEL` and `FWORDER` components with the `deviate-frequent` heuristics, the argument selection and the pairwise dominance models respectively. Thus after reporting the merit of each individual proposal, we now would like to see whether the same effect remains when we combine them together. In other words, we would like to understand whether the three proposals are orthogonal to each other as such the combination can produce additional performance gain. Similar to the previous chapters, we conduct the experiments in two scenarios: the perfect lexical choice and the full translation task, and dedicate a section to each scenario. Specific to the full translation task scenario, we compare the performance of the improved F W S model with the other two baseline models (Moses and Hiero). In analyzing the results, we use the statistical significance test, casual inspection on the translation output and the parameters size needed by the model. We end each section with a discussion of the results upon which our future work will be drawn.

9.1 Perfect Lexical Choice

In this scenario, we would like to study the effect of combining the three models (i.e. the `deviate-frequent` heuristic, the argument selection model and the pairwise dominance model) to the reordering quality and to the basic F W S model' errors each model designs to address. To study the effect of these models to the reordering quality, we use the BLEU score; while to study the effect of these models to the basic F W S model's error, we use the following intrinsic metrics: `pORI-acc`, `false-mono`, `unhandled-arg` and `pORD-acc`. For the `pORD-acc` and `pORI-acc`, the higher the score the better the performance is; while contrary for the `false-mono` and `unhandled-arg`, the lower the error the better the performance is.

As a recap, `pORI-acc` measures the system's accuracy in assigning the correct `pORI` value to the surrounding phrases. Meanwhile, `false-mono` refers to the number of `pORI-acc` errors that correspond to the system's false recommendation of monotone reordering, while `unhandled-arg` the number of `pORI-acc` errors that are due to the arguments beyond the function word's first neighbor. Finally, `pORD-acc` refers to the accuracy of assigning the `PORDER` predicate between two competing function words. Similar to the previous experiments, we consider all lexical items in the computation of the `pORI-acc, false-mono` and `unhandled-arg` metrics, while we consider the gold standard function words in the computation of the `pORD-acc` metric. Note that computing the ground truth for all these metrics requires manual word alignment.

Table 9.1 reports the results of our experiments, which all use the same number of lexical items ($N = 128$). In the table, *ori* represents the basic F W S model described in Chapter 5. Meanwhile, the subsequent three rows represent the basic F W S model with an improvement in one F W S component. In particular, *ori+δ=0.5* comes with the improvement in the `FWID` component, *ori+argsel_auto* the `ARGSEL` component, and *ori+dom+domp* the `FWORDER` component. The last row

System	BLEU	pORI-acc (#errors)	false-mono	unhandled-arg	pORD-acc
ori	79.58	78.89% (5,668)	3,245	2,080	76.68%
ori+δ=0.5	79.97	79.35% (5,543)	3,052	2,045	77.58%
ori+argsel_auto	80.35	79.52% (5,496)	3,418	1,975	77.63%
ori+dom+domp	80.90	79.89% (5,397)	3,395	1,929	78.72%
improved	81.57	80.76% (5,166)	2,996	1,869	80.05%

Table 9.1: Performance of the basic F W S model, the three proposals and the improved F W S models.

denoted as *improved* combines the improvement in all these three components.

Rows 2 to 4 shows that each model is doing a good job on its own task, bringing improvement to the specific component assigned. For instance, *ori+δ=0.5* gives the biggest error reduction in terms of the `false-mono` metric as compared to other two proposals. *ori+argsel_auto* also reduces the number of the `unhandled-arg` errors, although the reduction is less than the reduction of the `unhandled-arg` errors given by the *ori+dom+domp*. Meanwhile, the *ori+dom+domp* gives the best improvement in terms of `pORD-acc`.

When we look at the statistics of the *improved* model, we see a desirable result since it consistently gives the best results in all metrics, as shown in Table 9.3. For instance, the *improved* model reduces the number of `false-mono` error further from the best result given by *ori+δ=0.5*. Similarly, combining the dominance model with the `deviate-frequent` heuristic and the argument selection model increases the `pORD-acc` almost 1.50%, which doubles the increase given by the dominance model alone. The same trend also applies to the `unhandled-arg` error reduction, where the combination brings the error down further.

This incremental improvement translates to the increase in the `pORI-acc` as well as in the BLEU score. The combination of the three proposed models are able to produce a statistically significant improvement ($p < 0.01$) of 2.00 BLEU points

absolute over the baseline *ori* model. These experimental results suggest that the three improved models are orthogonal to each other, giving a complementary performance gain when combined.

While the improved F W S model is able to fix some of the basic F W S model's reordering errors, it still makes some errors, which still leaves some rooms for future improvement. We will elaborate our future work in the next chapter, but here, we identify some cases where further improvement is necessary.

In particular, the majority of the errors (`false-mono`=57.99%) still concern with the overly strong tendency toward recommending monotone reorderings. When we look at the orientation values of the function words that support non-monotone reordering, we notice that the probability mass for non-monotone reorderings is higher by only a small magnitude than the one for monotone reorderings. For instance, the orientation values of the most frequent word 的 (of) shown in Table 5.1 are divided almost equally between monotone and non-monotone reordering. This may suggest that contextual information beyond the function word is necessary.

9.2 Full Translation Task

Before comparing the improved F W S model with the state-of-the-arts phrase-based and syntax-based models, we are interested in evaluating whether the same incremental improvement observed in the perfect lexical choice scenario also applies to the full translation task scenario. Table 9.2 reports the BLEU score of the improved F W S model, in comparison with the basic F W S model and the three individual proposed models. Similar to the perfect lexical choice scenario, we report the improved F W S model with $N = 128$. As shown, the same incremental improvement is also evident in this scenario where the input word alignment is noisy.

System	BLEU
ori	24.92
ori+δ=0.5	25.29
ori+argsel_auto	25.59
ori+dom+domp	25.79
improved	26.45

Table 9.2: Performance of the basic and the improved F W S models along with the baseline models in terms of BLEU score.

Table 9.3 compares the improved and the basic F W S models with the state-of-the-art phrase-based and syntax-based models in terms of BLEU score. Similar to the Pharaoh system, we performed the minimum error rate training for Moses for different distortion limit (dl) settings, starting from 0 to 10 and report only the best result (dl=6).

As shown, the basic F W S model alone is able to outperform the two state-of-the-art phrase based models, although the improvement over the Moses system is not statistically significant. Meanwhile, the improved F W S model is able to consistently outperform all the baseline models but only a modest improvement over the Hiero system.

We then performed several analyses to highlight the benefit of our proposals. First, we examine the parameter size needed by each system. Table 9.3 provides such information. In reporting the parameter size, we produced not only the absolute size of the model but also its growth rate in terms of the maximum sentence length n and the lexical items N used. Note that we obtain the approximation for the baseline models from (Quirk and Menezes, 2006).

The lowest performing system, Pharaoh, only requires n^2 number of phrases, since the distortion penalty model requires no parameters. The state-of-the-art phrase-based models, Moses, requires n^2 additional space for storing the parameters of the unigram lexicalized model on top of the n^2 numbers of phrases. In terms

System	BLEU	p	p_+	p_0	p_-	Model size		
Pharaoh (dl=5)	22.44	< 0.01	1,112	291	597	phrases	=	213,336 (n^2)
						Total	=	213,336
Moses (dl=6)	24.87	< 0.01	582	1,316	102	phrases	=	213,336 (n^2)
						reordering	=	213,336 (n^2)
						Total	=	426,672
Hiero	26.08	> 0.05	759	545	696	rules	=	2,137,168 (n^6)
						Total	=	2,137,168
basic F W S	24.92	< 0.01	911	573	516	phrases	=	246,750 (n^2)[1]
						ori	=	29,929 ($\|Y\|$)
						Total	=	276,679
improved F W S	26.45	-	-	-	-	rules	=	246,750 (n^2)
						ori	=	29,929 ($\|Y\|$)[2]
						dom	=	34,917 ($\|Y\|^2$)
						domp	=	15,194 ($\|Y\|^2$)
						Total	=	326,790

Table 9.3: Performance of the basic and the improved F W S models along with the baseline models in terms of BLEU score. The statistical significance test measures the performance gain of the improved F W S model over the other models. p_+ refers to sentences where the improved F W S performs better, p_- refers to sentences where the improved F W S performs worse, while p_0 refers to sentences where the improved F W S performs equally well.

of the space requirement, Hiero is the highest, demanding the storage of rules which growth rate is in a high polynomial factor n^6. On the other hand, our F W S model only needs a modest space requirement. On top of the phrase translation table which size grows quadratically, the F W S model only requires space which size grows in a constant time with respect to the maximum sentence length, thus independent of the corpus size. We find these statistics to favor our F W S models, since it suggests that the models can achieve the state-of-the-art performance without introducing too many parameters that may expose the systems to data sparsity and over fitting problems.

As a final analysis, we are particularly interested in comparing the improved F W S model with the Hiero system – the strongest baseline system. In particular, we are interested in understanding the following two cases: 1) where the improved

F W S model outperforms the Hiero system and 2) where the Hiero system outperforms the F W S model. To do so, we look at sentences that belong to p_+ and p_-, where in p_+ refers to the first case while p_- to the second case. Table 9.3 shows the statistics. While the majority of performance differences are attributed to the lexical-related errors which are difficult to analyze, there are still some obvious reordering-related errors.

When we looked at the sentences that belong to p_+, we observed the adverse effect of the undergeneration and the overgeneration problems. Fig 9.1 illustrates Hiero's first type of mistakes, which concerns with the FSB model's low generalization power. As shown, the second rule after the root node (X_1) incorrectly swaps the neighboring phrases of a content word 查询 (query). In total, it incorrectly swaps the three-word phrase on the left and the four-word phrase on the right of the word. On the other hand, the improved F W S model is able to produce the correct translation by relying only on function words as anchors. This example also shows Hiero's vulnerability to the over fitting problem, since such a long-distance reordering is perhaps valid in some cases that are found in the training data but not to unseen cases.

Fig 9.2 illustrates Hiero's second type of mistakes, which concerns with the FSB model's undergeneration problem due to the non-adjacent nonterminal constraint. This constraint, as mentioned earlier, forbids Hiero from creating rules with adjacent nonterminals. Fig 9.2 basically shows a real case where this restriction is problematic. The construction in Fig 9.2 bears a close resemblance to the VP construction in Fig. 7.1 where the noun phrase, which is the second neighbor to the right of 为 (for), moves to the beginning of the VP. In some cases, Hiero can accommodate such VP constructions by remembering the actual wording of the noun phrase. Unfortunately, it cannot do so in this example because the noun phrase is not seen in the training data. On the other hand, the improved F W S model is

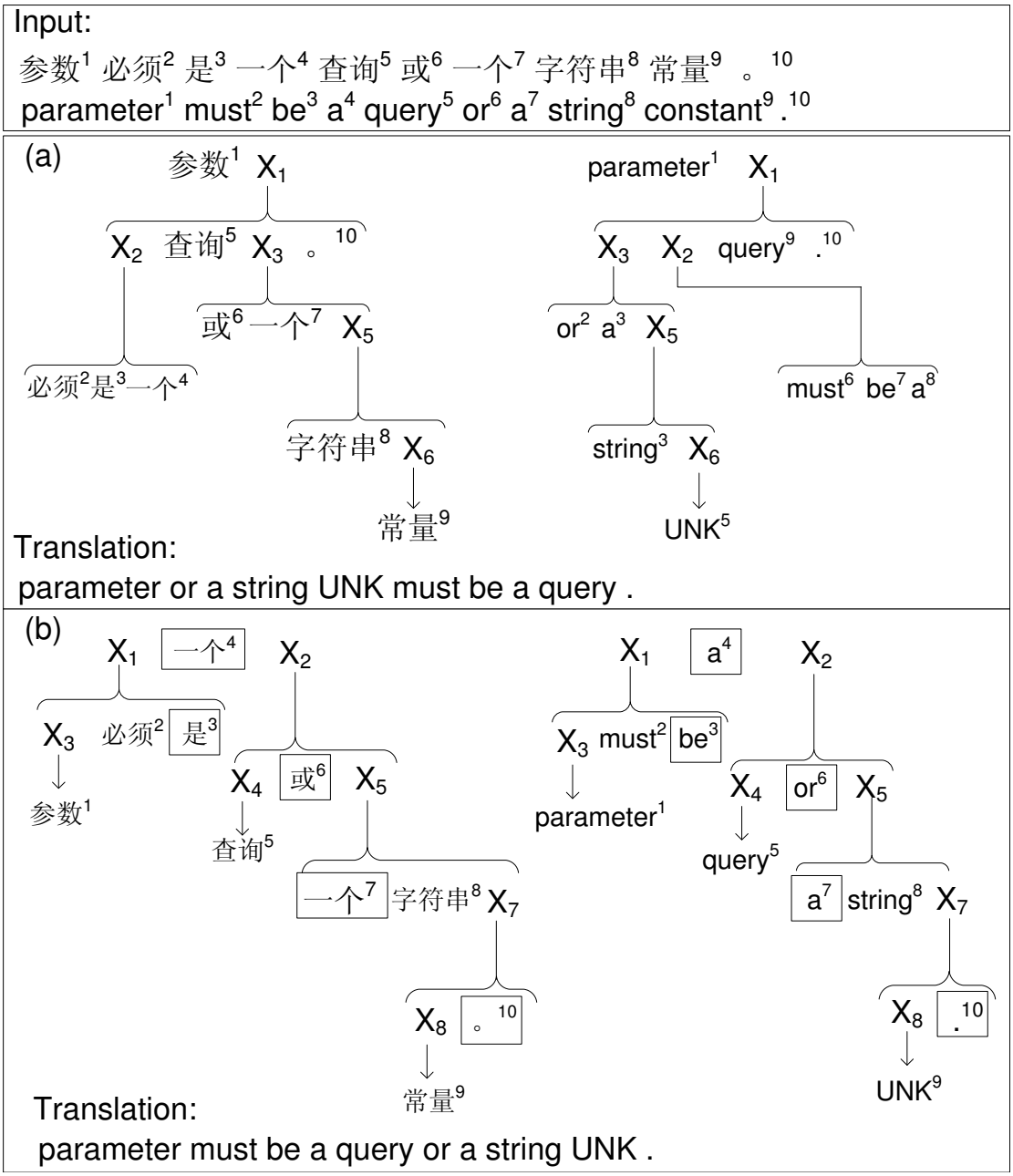

Figure 9.1: The first type of Hiero's mistakes that can be fixed by the improved F W S model. (a) shows the output of the Hiero system. (b) shows the output of the F W S system. The translation of each Chinese word is shown in the input box (the topmost box) as an English word having the same superscript with its Chinese counterpart.

able to handle this construction since it allows rules to take adjacent nonterminals.

Fig 9.3 illustrates Hiero's third type of mistakes, which concerns with the overgeneration problem in the FSB model due to ambiguous order of application. In particular, the figure shows that the ambiguous order of application causes Hiero to reorder an incorrect span of text. As shown, while Hiero is able to correctly predict that the left and the right nonterminals of the word 的 (genitive marker) must be swapped, it fails to recognize that the span of the right nonterminal must not include the comma. The correct reordering involves putting the comma at the level higher than the rules involving the word 的, such that the noun phrase is created first before joined with the rest of the text by the comma, as illustrated by the output of the F W S model in Fig 9.3b. In some cases, Hiero is able to fix this type of mistakes if in the training data, the comma appears somewhere after the marker 的. But again, this solution would involve the introduction of additional rules which may increase the risk of running into more severe over-fitting and data sparsity issues.

While the improved F W S model is superior to Hiero in some cases, we also observe some cases where Hiero performs better than the F W S model. Most of the reordering mistakes made by the F W S are apparently related to the VP construction similar to the one in Fig. 7.1, where the PP moves freely due to the flexibility of the English language. Fig. 9.4 shows the error and contrasts it with the output of the Hiero system which reordering is correct. As shown, while F W S is able to position the object after the verb, it fails to move the PP to the beginning of the sentence. Meanwhile, Hiero is able to do so since its hierarchical rules remember the context of the PP movement.

Theoretically, the F W S model should be able to accommodate such movements, for instance by treating the verb as the second neighbor of the preposition 中 (on) as that is what we observe in the model's output from the perfect lexi-

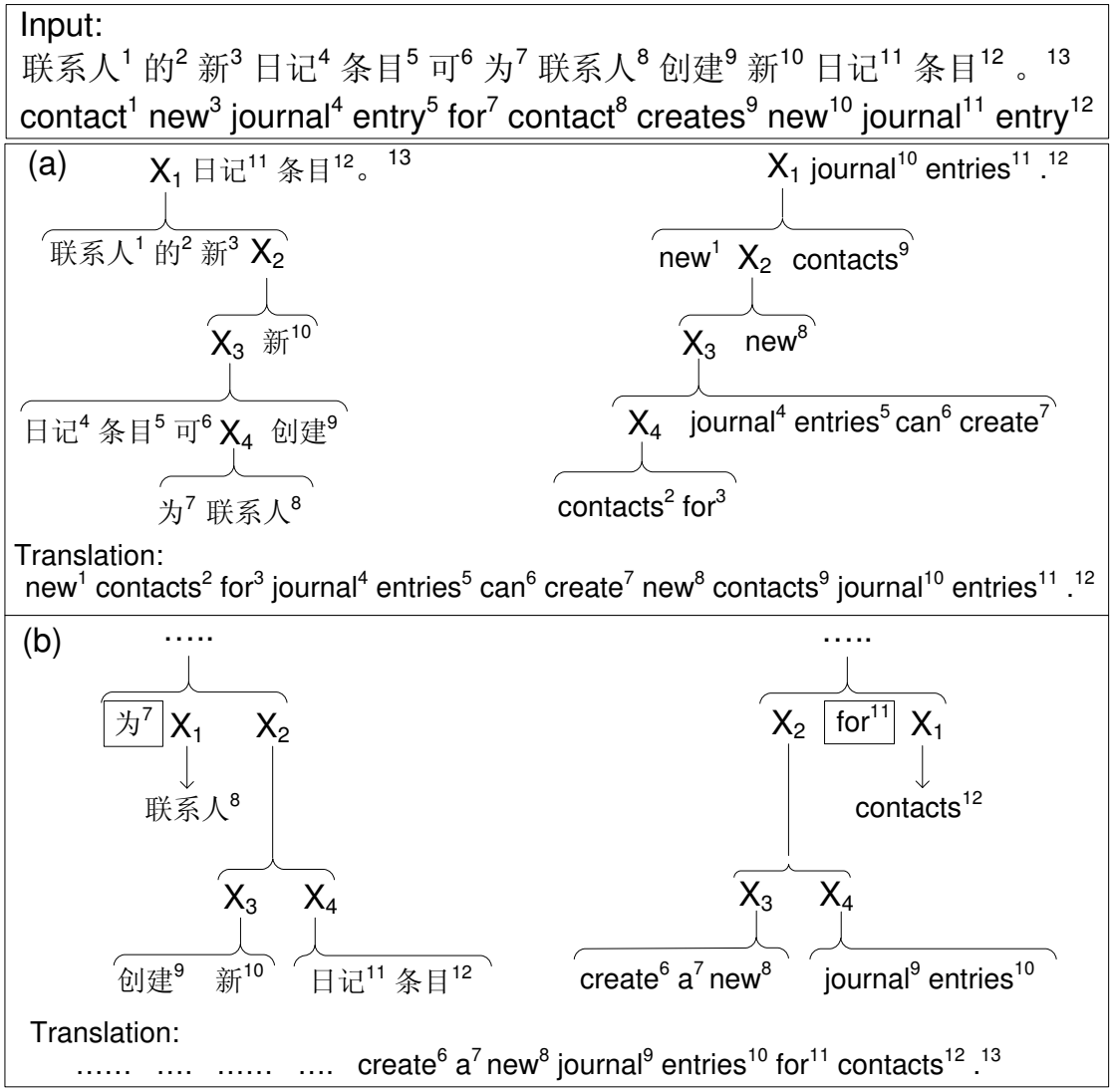

Figure 9.2: The second type of Hiero's mistakes which can be fixed by the improved F W S model. (a) shows the output of the Hiero system. (b) shows the output of the F W S system. The translation of each Chinese word is shown in the input box (the topmost box) as an English word having the same superscript with its Chinese counterpart.

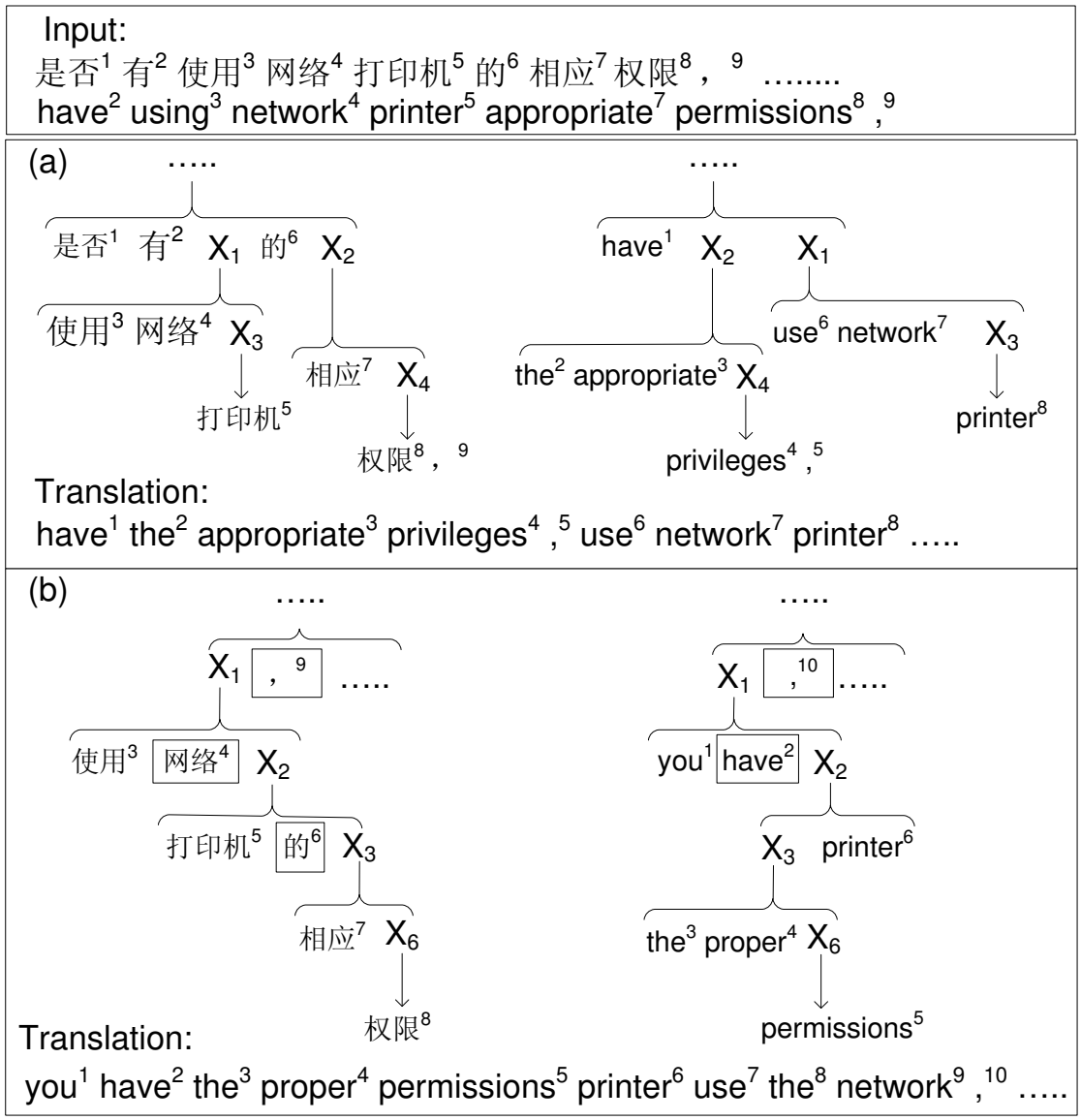

Figure 9.3: The third type of Hiero's mistakes which can be fixed by the improved F W S model. (a) shows the output of the Hiero system. (b) shows the output of the F W S system. The translation of each Chinese word is shown in the input box (the topmost box) as an English word having the same superscript with its Chinese counterpart.

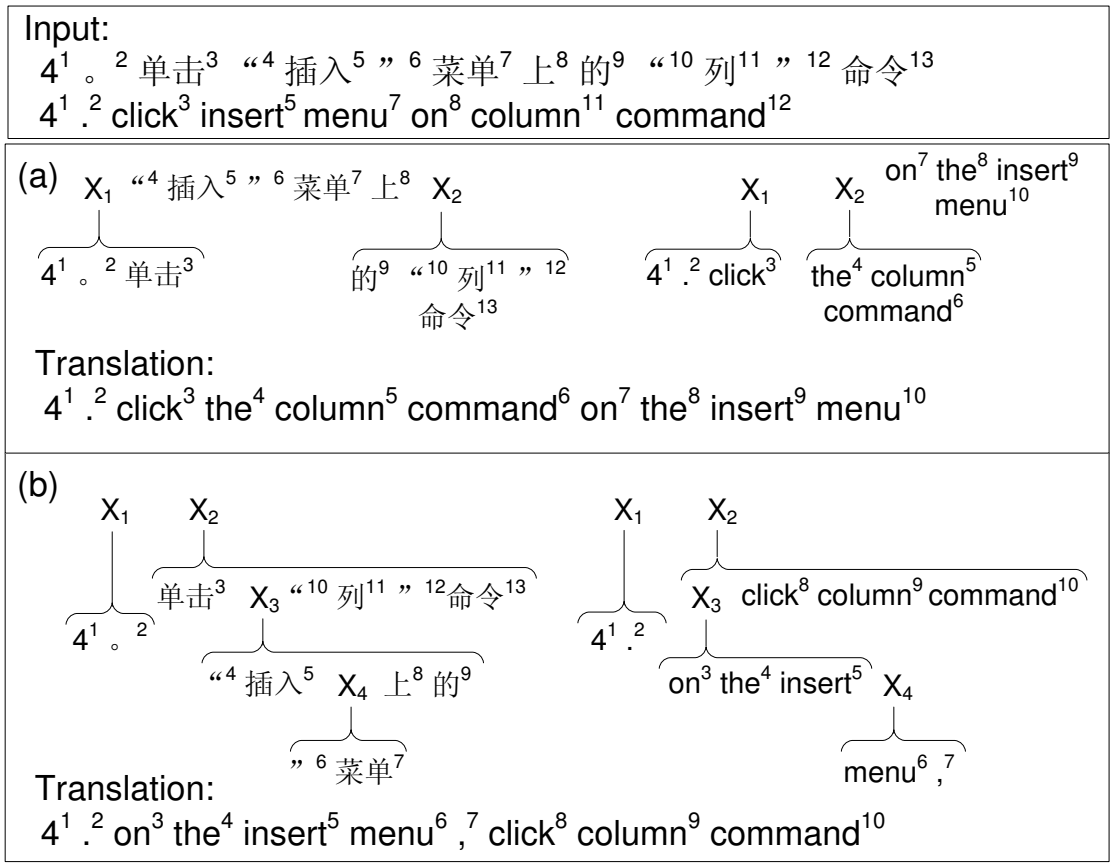

Figure 9.4: The mistake of the F W S model where the PP “插入” 菜单中(on the insert menu) should be moved to the beginning of the sentence. (a) shows the output of the F W S model. (b) shows the output of the Hiero system. The translation of each Chinese word is shown in the input box (the topmost box) as an English word having the same superscript with its Chinese counterpart.

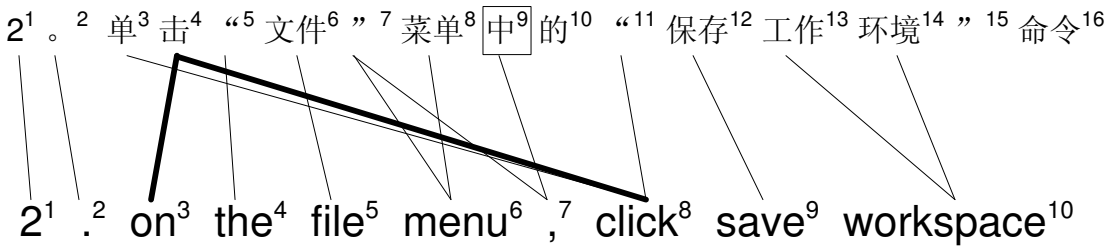

Figure 9.5: An illustration of the alignment error that can hamper the orientation model from learning its parameters. The Chinese character in the box represents the head, which the orientation model is trying to estimate. The thick lines represent the alignment errors that hamper the orientation model to learn the movement of the verb.

cal choice scenario. Furthermore, we also observe that such movements appear in a significant number in the training data, thus the statistics should strongly recommend the PP movement. When we carefully analyzed the training data, we found out that the underlying reason is the alignment error. Fig. 9.5 shows the automatically-obtained alignment of the sentence in Fig. 9.4 extracted from the training data. As shown, there are some alignment errors where the punctuation mark (“) is incorrectly aligned to the verb 单击 (click). We observed that such an error constitutes a large part of all the alignment errors.

Due to this error, our orientation model is unable to learn the movement of the second neighbor to the right of the preposition 中 (on), since the orientation model training requires the neighbor to be contiguous phrase translation. On the other hand, Hiero is able to learn hierarchical rules needed to correctly reorder the testing sentences from this training sentence since its grammar induction process is more robust to alignment errors. We hope that we can address this error in the future because we observed that they cause most of the reordering errors in this full translation task. Nonetheless, this analysis suggests that the two systems make different type of errors and that the two can complement each other to produce an even better result.

9.3 Summary

In this chapter, we have introduced the improved F W S model, which combines the three models developed earlier. We study the results and are pleased to see that the three models are orthogonal and combining them produces an incremental improvement in both reordering and overall translation quality. Furthermore, this conclusion applies not only in the perfect lexical choice scenario but also in the full translation task scenario. We have also compared the improved F W S model with the state-of-the-arts models, including the strongest model available as of this thesis writing. The experimental results suggest that the improved F W S model is able to outperform all the models – although only marginally over the strongest baseline – with the advantage of using much less number of parameters, which size is independent of the corpus size. We have also hinted some possible further improvement that can be pursued in the future from our analyses of the output.

Chapter 10

Adaptation to Hiero

In this chapter, we combine the strength of the state-of-the-art Hiero model and the our F W S approach, given the evidence in Section 9.2 that the two models are better than each other at orthogonal cases. In Section 10.1, we first discuss several modifications necessary to allow adapting the F W S approach into the Hiero model both in terms of the grammar formalism and the statistical models. In Section 10.2, we discuss the experimental setup, especially the data used in the experiments. In Section 10.3, we report the experiments and show empirically that adapting the F W S idea improves the state-of-the-art Hiero model in a large-scale experimental setup. In Section 10.4, we conclude with a summary.

10.1 Several Notes about Adaptation

The most striking difference between the F W S model and the Hiero model is that in the F W S model, function words always appear individually in a separate terminal rules, while in the Hiero model, they may appear together with other non-function words. Thus, adapting the F W S model requires dealing with this difference. Apparently, adapting the F W S approach into the Hiero model only

requires one small modification to the Hiero model's formalism, namely extending the correspondences between the source and target language strings to also include lexical items. Recall that in the Hiero model, the correspondences between the source and the target strings (represented by the $\sim$ symbol in Rule 1.1) are restricted only between nonterminals.

To make the discussion more concrete, let's consider the following Hiero rule as a case in point:

$$X \rightarrow \langle \text{电脑 和 } X,\ \text{computers and } X \rangle \tag{10.1}$$

As shown, the co-indexation is applied only on the Xs but doesn't involve the lexical items.

The first step to adapt the F W S idea involves extending the correspondences between the source and the target strings to also include the lexical items. Thus, after such an extension, Rule 10.1 would become:

$$X \rightarrow \langle \text{电脑}_1 \text{ 和}_2 X, \text{computers}_1 \text{ and}_2\ X \rangle \tag{10.2}$$

This extension is relatively straightforward, only involving an additional bookkeeping during the rule extraction process. To extract rules like Rule 10.2, we basically use the same rule extraction method as described in (Chiang, 2005) but instead of discarding the word-alignment information, we keep them. The extractor tools that come with the Hiero model already have the capabilities to keep the word-alignment information in the extracted rules.

Essentially, with such an extension, we can estimate the parameters of the F W S models even though function words are not treated exclusively. Since the Hiero and the F W S models share the same log-linear architecture, the upcoming adapted statistical models would act as features in the Hiero model along side standard Hiero features. We refer the reader to (Chiang, 2005) for the complete list of Hiero standard features.

Another important bit of detail in adapting F W S idea involves the lexicalization of nonterminals as the design of the Hiero model doesn't include lexicalization. To do so, we emulate lexicalization through the development of *stateful* features (similar to what we've done for the F W S model) that are already accommodated by the implementation of the Hiero model. Stateful features refer those features that require external information to compute their score, as opposed to *stateless* features that only use internal information. In stateful features, rules can carry and pass arbitrary information into another rules in addition to the current score.

An example of stateful feature is the target n-gram language model, which computation requires the context of n-1 words from the children rules. Thus, the application of each rule produces not only the target language model score but also the n-1 words context that will be used for the computation of the target language model of other rules. Meanwhile, examples of stateless features are standard phrase-based features like translation probability or lexical weight, which scores are fixed regardless of the context of the rules.

10.1.1 Adapting Orientation Model

As a recap, the orientation model evaluates the reordering of phrases (Xs) with respect to their neighboring function words (Ys) through `pORI`(X, Y) function that outputs one of the four orientation value given a particular X and a particular Y. In the F W S model, the orientation value of a phrase can be evaluated directly because there is a special treatment for those words that belong to function words, i.e. function words always appear in individual units. In the Hiero model, function words sometimes have been embedded inside the rule and treated like any other lexical items as shown in the above example rules – 和 (and) is a function word but 电脑 (computers) is not.

In the original Hiero rules, it is often impossible to determine the orientation values of neighboring phrases of a function word. Let's take Rule 10.1 as a case in point. Identifying that 和 (and) is a function word is as easy as enumerating all the source words in the rules and check whether any word belongs to the function word list. However, estimating the arguments is non-trivial as they can span more than one word. And even though the arguments can be perfectly estimated, estimating their orientation is non-trivial since there is no information about where these arguments end up in the target side, e.g. there is no information whether 电脑 is translated to "computers" or "and".

However in contrast, the orientation model score can be calculated straightforwardly in the Hiero rules extended with word-alignment information like Rule 10.2. In principle, we apply the parameter estimation procedure which we described in Section 5.3 to extract the orientation value of neighboring phrases of a function word. With the word-alignment information (Rule 10.2), we can estimate that the orientation value of the left neighbor of 和 (and) is Monotone Adjacent (MA). Note that since the right argument of 和 (and) is a nonterminal X, we delay adding the orientation model score for that argument until a concrete phrase substitutes the nonterminal.

In the Hiero model, the orientation model is a stateful feature which requires information about the function words and the word-alignment to be propagated up to the structure. Since we focus only on evaluating the orientation model of the neighboring phrases, we only propagate the information about word-alignment and the information about those function words whose left and right arguments haven't been scored. Thus, once a concrete phrase has substituted the X in Rule 10.2 and the orientation score for the right argument of 和 (and) has been computed, Rule 10.2 doesn't propagate the information about 和 (and) and the word alignment anymore.

10.1.2 Adapting Pairwise Dominance Model

The role of the pairwise dominance model is resolve the order of rule application based on the word-alignment between two function words that are the heads of the competing rules. In this model, the word-alignment information is essential to determine the dominance value between two function words. In short, the estimation of the dominance value shares the same principle as the orientation value. In the pairwise dominance model, we attach the information about all function words with the corresponding word-alignment, except those function words whose left and right dominance values have been scored.

10.1.3 Adapting Function Word Identification Method

The function word identification technique developed for the F W S model requires no modification when adapted to the Hiero model as it is performed as a preprocessing step.

10.1.4 (Not) Adapting Argument Selection Model (Yet)

The role of argument selection model is to select an appropriate set of arguments for a particular function word based on how likely the arguments to move when translated. In F W S model, the argument selection model uses the idea of head-outward process similar to the Collins parsing model (Collins, 2003). Adapting the head-outward process modeling in the Hiero model setting is unfortunately non-trivial especially because function words are often embedded in the middle of the rules. For this particular reason, we reserve the adaptation to the argument selection model for future work.

10.2 Experimental Setup

We tested the effect of adapting our function word-based reordering idea on Chinese-to-English translation task. Similar to our previous experiments, we report performance using the BLEU score, and assess the statistical significance of the results of our experiments using the standard bootstrapping approach (Koehn, 2004b). Following the best result in the previous chapters, we equate function words as the $N = 128$ most frequent words in the corpus.

We trained the system on the NIST MT06 Eval corpus excluding the UN data (approximately 900K sentence pairs). For the language model, we used a 5-gram model with modified Kneser-Ney smoothing (Kneser and Ney, 1995) trained on the English side of our training data as well as the whole portion Gigaword v2 English corpus. We used the NIST MT03 test set as the development set for optimizing interpolation weights using minimum error rate training (MERT). We carried out evaluation of the systems on the NIST 2006 evaluation test (MT06) and the NIST 2008 evaluation test (MT08). We segmented Chinese as a preprocessing step using the segmenter from Harbin Institute of Technology (Zhao et al., 2001). As for the Hiero model, we are grateful to the model's author which provide all the necessary tools including the decoder and the scripts to extract rules from parallel text.

10.3 Results

Table 10.1 reports the result of our incremental experiments. We start with a baseline experiment to evaluate the performance Hiero system without employing any adapted model and then incrementally add new adapted models one at a time before adding them all together. As shown in the first three rows of Table 10.1, adding adapted models – both the orientation model ($+ori$) and the pairwise dom-

System	MT06	MT08
Hiero	29.29	22.15
+ori	*30.14*	22.84
+dom	*30.15*	*23.10*
+ori+δ=0.5	*30.59*	*23.50*
+dom+δ=0.5	*30.84*	*23.56*
+ori+dom+δ=0.5	**31.10**	**23.98**

Table 10.1: Performance of the baseline Hiero model and the Hiero model employing adapted F W S model in terms of BLEU score. Systems' performance that give statistically significant improvement over the baseline Hiero model are in *italics* while those that give the best performance are in **bold**.

inance model (*+dom*) – gives statistically significant gain in both MT06 and MT08 sets – except a not statistically significant but notable improvement given by the orientation model in MT08 set.

In rows *+ori+δ=0.5* and *+dom+δ=0.5*, we use the *deviate-frequent* heuristic to construct the function word list, instead of simply equating function words as the top 128 most frequent words in the corpus. Note that the number of lexical items in this set of experiments is still 128. As shown in these two rows, improving the quality of the function word list leads to a significant incremental gain. This gain is consistent for both orientation and pairwise dominance models as well as across the MT06 and MT08 sets.

The final row *+ori+dom+δ=0.5* shows the performance of employing all adapted models into the Hiero model. We are pleased to see the results as employing all adapted models together provides an incremental gain, consistent with the experiments with the F W S model, scaling the F W S approach to large-scale experiments.

10.4 Summary

In this chapter, we focus on adapting our function word idea into the state-of-the-art Hiero model and show the benefit of this idea beyond the framework we develop in this thesis. We show that some of the models developed for the F W S model can be adapted into the Hiero model by extending Hiero rules to include the word-alignment information. We show the virtue of our function word-based reordering idea in improving the performance of the state-of-the-art model in a large-scale experiment.

Chapter 11

Conclusion

The research presented in this thesis identifies weaknesses of the current approaches to the reordering task in the context of Statistical Machine Translation (SMT) and offers both theoretical and implemented solutions to address them. In this chapter, we summarize the main research contributions of this work, then list the main limitations of this work, discuss future research directions and conclude with implications of this work on the field of SMT as a whole.

11.1 Main Contributions

The main contribution of this thesis is in the proposal of using function words as the anchor to guide the reordering process. Function words are linguistically vital in explaining the grammatical relationship among phrases within a sentence and projecting them together with their dependant arguments to another language often results in structural changes to the realized sentence.

In this thesis, we have developed this idea in the context of the syntax-based approach, referred to as the Function Words, Syntax-based (F W S) approach. In a nutshell, the characteristics of the F W S approach are as follows: 1) it comes

with two types of nonterminal: function words and arguments; 2) it lexicalizes nonterminals with function words; and 3) it models the generation of nonterminals as a head-outward process. We exploited these characteristics to better address the undergeneration and the overgeneration problems that are found in the existing formally syntax-based models.

Under such a knowledge-poor environment, formally syntax-based models approximate their rewrite rules from parallel texts which provide no structural information. Without such knowledge, these models typically rely on a combination of one generic nonterminal symbol, lexical items and some heuristics. While the combination of these three features provides state-of-the-art performances, they make the models susceptible to both the undergeneration and the overgeneration problems.

As has been demonstrated in this thesis, the F W S approach is able to alleviate both the undergeneration and the overgeneration problems. Instead of relying on lexical items of any type, it relies on heads which are equated to function words. In practice, function words correspond to a small, fixed set of lexical items, making our approach not only linguistically-grounded but also relatively compact. But more importantly, it makes the model scalable to incorporate more information to provide a stronger structural preference. In our experiments, we show that our proposed model outperforms the currently available statistical systems and the performance gain is statistically significant.

Concretely, we use the function word idea to make the following contributions:

- The function word identification method.
- The argument selection model.
- The pairwise dominance model.

We elaborate these three contributions individually in the following sections.

11.1.1 The function word identification method

One important aspect of the F W S approach is the identification of function words. The successful identification of function words ensures the success of the downstream process in the F W S approach. One contribution of this thesis is on the identification of function words. Specifically, we look at two easy to obtain statistics, namely: the frequency and the deviation statistics. The former refers to how frequent the word appear in the training data while the latter refers to how likely is the word's surrounding phrases to move. Thus, according to these statistics, we classify words as function words if they appear with high frequency and their surrounding words tend to move. We showed in our experiments in Chapter 6 that we can obtain a high quality list of function words that can improve the reordering quality.

We can also extend this idea beyond the F W S model. For instance, it can be used to extend the Hiero system or even the phrase-based system to identify the lexical items that give the most benefit to the reordering task. Or, we can also use this simple idea to evaluate the usefulness of modeling different level of abstraction. For instance, we can use this simple idea to decide whether a noun should be modeled at the lexical level or at a more abstract level (a noun class or a singular noun class).

11.1.2 The argument selection model

Another innovation of this thesis is the development of the argument selection model. Specifically, we allow a function word to have a flexible set of arguments, i.e. not restricted to the neighboring text; and use the statistics about where those arguments are likely to move to select the most appropriate set of arguments. This

model removes the practical restriction imposed by the state-of-the-art syntax-based models that forbid the creation of rules with adjacent nonterminals. In other words, we prefer to accommodate more arguments but at the same time treat the ambiguity by statistical means. We showed in our experiments that allowing a more flexible arguments coupled with our argument selection model provides a significant improvement gain.

This model can also be applied to other approaches. For example, it can be used by the Hiero system to promote judicious uses of adjacent nonterminals, alleviating its undergeneration problem. It can also be applied by a phrase-based system to allow more flexible context modeling, beyond just the preceding phrase.

11.1.3 The pairwise dominance model

The third key contribution of this work is in the development of the pairwise dominance model. Under a knowledge-poor environment where no structural information is available, we developed this model to approximate the order of rule application by looking at the phrase alignment around every neighboring function words. We exploit the fact that different order of application produces different kind of phrase alignment around the function words. We showed in the experiments that our pairwise dominance model is able to give a significant improvement gain.

Again, this model is not restricted to the F W S approach but also to other models, such as the Hiero system by perhaps extending the definition of head beyond the function word class, although one may have to be careful with data sparsity issue. It can also be applied as a phrase-based model since this model is approximated only via phrase alignment.

11.2 Limitations and Future Work

The development of the function word idea takes the reordering process a step further. However, we acknowledge some limitations in the current implementation. We examine these obstacles and make recommendations for future research that can address these issues.

- **The applicability of the idea of function words** - Function words have a significant grammatical role in analytical languages, such as Chinese and English, where the syntactic and the semantic of the sentence are shaped by the use of function words and embedded in the word order. However, this idea may not be directly generalizable to heavily agglutinative languages, such as Arabic, where the grammatical marker is attached to the semantic unit with the use of affixes. We suspect that while function word centric reordering has its linguistic ground in analytical languages, it requires some adaptation prior to its implementation to agglutinative ones. To accommodate such languages, the F W S approach probably has to go to a more finer-grained analysis, i.e. morpheme units, by first performing morphological analyses to the source sentence.

- **The knowledge about argument boundary** - In this thesis, we have experimented with a simple solution to approximate the argument boundary knowledge using a shallow linguistic analysis based on text chunking. While text chunking is good for bracketing the monolingual text, we found the output is not suitable for our purpose of reordering phrases. It is often the case that the phrase boundary of an argument in one language does not agree with its projection in another language. Although we can achieve a considerable good performance without the proper knowledge about the argument boundary, we still see some obvious errors in the translation output that are directly

attributed to the absence of such knowledge.

- **The single function word head** - In this thesis, we restrict the definition of function words to single words. In many languages, there are cases where one function word is not independent of each other, such as in the case of split function words: 从...上 which means "from above" in English. Currently, the F W S approach has yet to cater these function words, treating them as two separate entities. We observe through a casual inspection that some of the mistakes made by our system are due to these cases.

Aside from the limitations of the function words-centric approach, there are some natural extensions of this thesis in the direction of future research. Here, we look briefly at several routes of future research:

- **Enriching the representation of function words** - Currently, the head-driven SCFG only captures the reordering that is influenced by a single head. We suspect that we can improve the performance of the F W S approach further by enriching the heads with finer-grained information. One simple way to enrich the function words is to complement the lexical information with empirical evidence such as the position of the function words in the sentence. Another way is by coupling two neighboring function words together, as doing so may suggest a more precise reordering. For example, the orientation statistics for the most frequent word 的 (of) give almost equal probability to monotone and reverse reordering - with a little more probability mass to the latter. Although it is useful in practice, this part of the model contains high entropy, thus requires other components to add the additional discriminating power. We often can find more refined statistics when the word appears next to another function word. For instance, we observe that when 的 (of) appears next to 上 (on), it is most likely to swap the surrounding text. How-

ever, when 的 (of) appears next to other heads, it is most likely to suggest monotone reordering.

- **Enriching the representation of arguments** - Currently, little information about arguments are involved in the reordering process. The F W S model only uses positional information such as the argument's location with respect to the head. When reordered, the function words treat all arguments similarly in different context. Obviously, extra information about the arguments would be beneficial. For instance, the second neighbor argument of the function word 为 (for) should be restricted only to verbal phrases for the VP construction illustrated in Fig. 7.1. In the future, we hope to explore different methods to exploit the evidence supplied by the arguments.

- **Going beyond two labels** - The previous two routes can be seen as introducing a new set of nonterminal labels into the grammar. If we relate the syntax-based approach to the monolingual grammar induction process, the introduction of function words represents the first effort to induce the complete set of word classes from raw text. In the future, we hope to benefit from the more mature field of monolingual grammar induction, particularly to mimic a typical road map taken to induce the grammar in an unsupervised way. As such, the resulting grammar contains richer information that encodes stronger structural preferences.

- **Moving to the knowledge-rich environment** - We intend to integrate the word class information (perhaps in terms of POS tag or lexical categories) into our framework, which is similar in spirit with the previous route of future work but here the knowledge source comes from linguistic annotation. The additional layer information allow us to generalize the heads into a more coarse-grained tokens and to abstract away from the arguments' lexical items.

The latter is extremely important as the generalized representation of the arguments are still missing in the current implementation.

- **Extending the idea to the full translation task** - Although we show that our model performs well in the full translation task, we believe that we have only touched the surface benefit of the F W S approach, especially when we observed that alignment errors intrude and hamper the full realization of this idea. In the future, we hope to better scale up this approach to the full translation task, which may include a better proposal for alignment algorithms that is geared toward function word modeling. Another possible route is to integrate some ideas from hierarchical phrases into our framework, especially to make our model more robust to alignment errors.

11.3 Revisiting the Syntax-based Approach

The move to the syntax-based approach has since brought SMT research closer to natural language formalism. However, two inter-related open research questions arise as to what is the appropriate representation to model the structural difference between the source and target languages and how to estimate the parameters of such a representation. As of this thesis writing, there is no consensus about how to answer these two questions. The formally syntax-based approach strives for portability, designing a system which can be adopted to a new language pair with little effort. However, the structural difference is only represented by a single generic nonterminal symbol coupled with lexical items. On the other hand, the linguistically syntax-based approach strives for fidelity, designing the system to be faithful to the linguistic annotation prepared by human linguists. In practice, such a system has to work on an environment that is far from ideal where noise interfere. Besides, it is unclear whether such linguistic annotation provides a suitable level of

representation for the task – the same issue that also arises in the monolingual parsing task. We view both approaches as two efforts that start from different starting points, approaching the ideal syntax-based model somewhere in the middle.

In this thesis, we have engaged ourselves to seek the answer to these questions when we touch on a specific subtask of the translation task: the reordering task. Through the development of the function word idea, we hypothesize that the fact that function words provide the essential syntactic information is beneficial for reordering. We demonstrate the utility of such approach in the formally syntax-based approach, where no linguistic annotation is available, but the identity of function words is identifiable. We see our thesis as the one that brings formally syntax-based approach one step closer to the ideal syntax-based model. It is our hope that we have also provided some useful ideas about what does and what does not work in this framework.

References

Abney, Steven. 1987. *The English Noun Phrase in Its Sentential Aspect*. Ph.D. dissertation, Department of Linguistics, MIT, Cambridge, Massachusetts.

Aho, A. V. and J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3(1):37–56, February.

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

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

Berger, A., P. Brown S. Della Pietra V. Della Pietra A. Kehler R. Mercer. 1996. Language translation apparatus and method using context-based translation models, U.S. Patent 5,510,981.

Birch, Alexandra, Miles Osborne, and Philipp Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16, Prague, Czech Republic, June. Association for Computational Linguistics.

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

Chan, Yee Seng, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th An-*

nual Meeting of the Association of Computational Linguistics, pages 33–40, Prague, Czech Republic, June. Association for Computational Linguistics.

Chen, Wenliang, Yujie Zhang, and Hitoshi Isahara. 2006. An empirical study of chinese chunking. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 97–104, Sydney, Australia, July. Association for Computational Linguistics.

Cherry, Colin and Dekang Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 105–112, Sydney, Australia, July. Association for Computational Linguistics.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Chino, Naoko. 2001. *All about Particles: a Handbook of Japanese Function Words*. Tokyo: Kodansha International.

Cocke, John. 1969. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University.

Collins, Michael. 2003. Head-Driven statistical models for Natural Language parsing. *Computational Linguistics*, 29(4).

Collins, Michael, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Cowan, Brooke, Ivona Kučerová, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 232–241, Sydney, Australia, July. Association for Computational Linguistics.

Dempster, A. P., N. M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

DeNero, John and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.

Deyi Xiong, Min Zhang, Ai Ti Aw Haitao Mi Qun Liu and Shouxun Lin. 2008. Refinements in btg-based statistical machine translation. In *Proceedings of IJCNLP 2008*, Hyderabad, India.

Fox, Heidi. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 304–311, Philadelphia, July.

Galley, Michel, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Germann, Ulrich. 2003. Greedy decoding for statistical Machine Translation in almost linear time. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 72–79, Edmonton, Alberta, Canada, May. Association for Computational Linguistics.

Germann, Ulrich, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada.

2001. Fast decoding and optimal decoding for machine translation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 228–235, Toulouse, France, July. Association for Computational Linguistics.

Green, T.R.G. 1979. The Necessity of Syntactic Markers: Two Experiments with Artificial Languages. *Journal of Verbal Learning and Behavior*, 18(4):39–71.

Harris, Zellig S. 1954. Distributional structure. *Word*, 10(23):146–162.

Howard, Jiaying. 2002. *A Student Handbook for Chinese Function Words*. The Chinese University Press.

Kasami, Tadao. 1963. An efficient recognition and syntax analysis algorithm for context-free languages. Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, MA.

Kneser, R. and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing95*, pages 181–184, Detroit, MI, May.

Knight, Kevin. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

Knight, Kevin and Jonathan Graehl. 2005. An overview of probabilistic tree transducers for natural language processing. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing CICLing*, volume 3406 of *Lecture Notes in Computer Science*, pages 1–24. Springer.

Koehn, Philipp. 2004a. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In Robert E. Frederking and Kathryn Taylor, editors, *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas*, volume 3265 of *Lecture Notes in Computer Science*, pages 115–124. Springer.

Koehn, Philipp. 2004b. Statistical significance tests for machine translation evaluation. In *Proceedings of Empirical Methods in Natural Language Processing 2004*, pages 388–395, Barcelona, Spain, July.

Koehn, Philipp, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of The International Workshop on Spoken Language Translation 2005*.

Koehn, Philipp and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation, June.

Koehn, Philipp and Christof Monz. 2005. Shared task: Statistical machine translation between European languages. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 119–124, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Koehn, Philipp and Christof Monz. 2006. Manual and automatic evaluation of machine translation between European languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City, June. Association for Computational Linguistics.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational*

Linguistics, pages 127–133, Edmonton, Alberta, Canada, May. Association for Computational Linguistics.

Kumar, Shankar and William Byrne. 2005. Local phrase reordering models for statistical machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 161–168, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Liang, Percy, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia, July. Association for Computational Linguistics.

Liang, Percy and Dan Klein. 2008. Analyzing the errors of unsupervised learning. In *Proceedings of ACL-08: HLT*, pages 879–887, Columbus, Ohio, June. Association for Computational Linguistics.

Liu, Yang, Yun Huang, Qun Liu, and Shouxun Lin. 2007. Forest-to-string statistical translation rules. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 704–711, Prague, Czech Republic, June. Association for Computational Linguistics.

Liu, Yang, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.

Marcu, Daniel, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Lan-*

guage Processing, pages 44–52, Sydney, Australia, July. Association for Computational Linguistics.

Menezes, Arul and Chris Quirk. 2007. Using dependency order templates to improve generality in translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

Nagata, Masaaki, Kuniko Saito, Kazuhide Yamamoto, and Kazuteru Ohashi. 2006. A clustered global phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 713–720, Sydney, Australia, July. Association for Computational Linguistics.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Och, Franz Josef and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, Franz Josef and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Och, Franz Josef, Nicola Ueffing, and Hermann Ney. 2001. An efficient A* search algorithm for statistical machine translation. In *Proceedings of the ACL*

2001 Workshop on Data-Driven Methods in Machine Translation, pages 55–62, Tolouse, France.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Quirk, Chris and Arul Menezes. 2006. Do we need phrases? Challenging the conventional wisdom in statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 9–16, New York City, USA, June. Association for Computational Linguistics.

Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Radford, Andrew. 1998. *Transformational Grammar*. Cambridge University Press, Cambridge.

Rambow, Owen, Bonnie Dorr, David Farwell, Rebecca Green, Nizar Habash, Stephen Helmreich, Eduard Hovy, Lori Levin, Carnegie Keith J. Miller, Teruko Mitamura, Florence Reeder, and Advaith Siddharthan. 2006. Parallel syntactic annotation of multiple languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Setiawan, Hendra, Min-Yen Kan, and Haizhou Li. 2007. Ordering phrases with function words. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 712–719, Prague, Czech Republic, June. Association for Computational Linguistics.

Stolcke, Andreas. 2002. SRILM — An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing, Volume 2*, pages 901 – 904, Jun.

Tillman, Christoph. 2004. A unigram orientation model for statistical machine translation. In *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Tillmann, Christoph and Tong Zhang. 2005. A localized prediction model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 557–564, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Tillmann, Christoph and Tong Zhang. 2007. A block bigram prediction model for statistical machine translation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(3).

Toutanova, Kristina, H. Tolga Ilhan, and Christopher D. Manning. 2004. Extensions to HMM-based statistical word alignment models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 87–94, Philadelphia, USA, Jul.

Vogel, S., Y. Zhang, F. Huang, A. Tribble, A. Venugopal, B. Zhao, and A. Waibel. 2003. The CMU statistical machine translation system. In *Proceedings of MT-Summit IX*, LA, USA, Sep.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen.

Wang, Chao, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint*

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 737–745.

Weaver, Warren. 1955. Translation (1949). *Machine Translation of Language*.

Wellington, Benjamin, Sonjia Waxmonsky, and I. Dan Melamed. 2006. Empirical lower bounds on the complexity of translational equivalence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 977–984, Sydney, Australia, July. Association for Computational Linguistics.

Wu, Dekai. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, Sep.

Wu, Dekai, Marine Carpuat, and Yihai Shen. 2006. Inversion transduction grammar coverage of Arabic-English word alignment for tree-structured statistical machine translation. In *IEEE/ACL 2006 Workshop on Spoken Language Technology (SLT 2006)*.

Xiong, Deyi, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia, July. Association for Computational Linguistics.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July. Association for Computational Linguistics.

Younger, D. 1967. Recognition and parsing of context free languages in time n3. *Information and Control*, 10:189–208.

Zens, Richard and Hermann Ney. 2003. A comparative study on reordering constraints in statistical machine translation. In *Proceedings of the 41st Annual*

Meeting of the Association for Computational Linguistics, pages 144–151, Sapporo, Japan, July. Association for Computational Linguistics.

Zens, Richard and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation*, pages 55–63, New York City, NY, June. Association for Computational Linguistics.

Zhang, Min, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, Ohio, June. Association for Computational Linguistics.

Zhang, Min, Hongfei Jiang, Aiti Aw, Jun Sun, Sheng Li, , and Tan Chew Lim. 2007. A tree-to-tree alignment-based model for statistical machine translation. In *Proceedings of Machine Translation Summit XI*, pages 535–542.

Zhao, Tiejun, Yajuan Lv, Jianmin Yao, Hao Yu, Muyun Yang, and Fang Liu. 2001. Increasing accuracy of chinese segmentation with strategy of multi-step processing. *Journal of Chinese Information Processing (Chinese Version)*, 1:13–18.

Zollmann, Andreas and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June. Association for Computational Linguistics.

Appendix A

Decoding Algorithm

We describe the decoding algorithm used by the function word, syntax-based (F W S) model to reorder source sentences. In essence, we employ the Cocke-Younger-Kasami (CYK) algorithm (Cocke, 1969; Younger, 1967; Kasami, 1963) to produce the translation given the source sentence F = f_1,f_2,...,f_J. We show the pseudo code of the algorithm in Alg. 1 and describe the most relevant details in subsequent sections.

Algorithm 1 function CYK(F: e_1^J, P: phrase translation table, M: models) → T^* (the best parse tree)

```
1: for start=0 to J-2 do do
2:    for end=start+1 to J do do
3:       chart[start,end].insert(initialize(F,start,end,P,M))
4:    end for
5: end for
6: for |span|=2 to J do do
7:    for start=0 to (J-|span|) do
8:       for mid=start+1 to start+|span|-1 do do
9:          chart[start,end].insert(merge(chart[start,mid],chart[mid,end],M))
10:      end for
11:   end for
12: end for
13: return best(chart[0,length])
```

A.1 The item and chart data types

The main data type in the algorithm is the item data type which holds the information about a node in a parse tree. Table A.1 lists the elements of the item data type.

Variables	Description
idx_1:integer	starting index
idx_2:integer	ending index
lbranch:item	left child
rbranch:item	right child
prob:double	probability score
$type \in \{X,\ Y,\ XY,\ YX\}$	node type
$op \in \{mono, rev\}$	operation type

Table A.1: A partial list of the variables and their descriptions of the item data type

The item's starting and ending indices (idx_1 and idx_2 respectively) refer to the white space index instead of the word index. For instance, the first word f_1 is represented by an item which starting and ending indices are 0 and 1 respectively, while the last word f_J by an item which starting and ending indices are $J-1$ and J respectively.

The node *type* is used to indicate the terminal rules' label (in cases of X and Y values) or to flag the partial expansion of a rule of rank three (in cases of XY and YX values). We need the latter to emulate the rules of rank three, since the CYK algorithm only creates a binary branching structure. The operation type indicates the reordering operation that is performed upon the *lbranch* and the *rbranch* children. This operation will affect the target language side of the node, i.e. whether the *lbranch* will be rewritten before or after in cases of monotone (*mono*) or reverse (*rev*) reordering, respectively.

Meanwhile, the chart data type is basically a strictly upper triangular matrix

which index starts from 0 and ends at J. Each element of the chart contains a list, which stores a collection of nodes of the same span. The insert() routine ensures that all the items in the list are sorted according to the item's probability score. In the exact implementation, we restrict the number of nodes kept in each sorted list and discard the others that fall beyond a certain threshold.

A.2 The initialize() routine

The initialize() routine prepares the chart data type by filling in the leaf nodes that are created from the entries of the phrase translation table. Similar to some variants of the phrase-based approach (such as the alignment template approach (Och and Ney, 2004) or those that use alignment constellation features (Liang et al., 2006)), we retain word alignment information for each phrase translation. This information is essential for the pairwise dominance model, especially for the estimation of the pORD predicate.

The initialize() routine basically enumerates all entries in the phrase translation table and performs the following operations:

- Checks whether an entry occupies a certain span in the source sentence. If it indeed occupies a certain span, then an item is created. The variables idx_1 and idx_2 are initialized with the span's starting and ending indices.

- Checks whether the newly created item belongs to either of the four item *type*. Specifically, it checks the entry's bordering word. It assigns X type if neither of the entry's bordering words is a head or Y if the entry contains only one word and it is a head. Meanwhile, it assigns XY if the ending word is a head, or YX if the starting word is a head. In cases where both the starting and the ending word belong to the head class, it creates two items: one item of XY type and another one of YX type.

- Determines the *op* variable for each newly created item, if the type of the newly created item is either XY and YX. Note that this step requires the information about the word alignment.

- Initializes the *prob* score with the language model score according to the model (M).

A.3 The merge() routine

This routine forms the main body of the decoding algorithm. Given two items of smaller span X_1 and X_2, the merge routine creates a new node by joining the two smaller nodes.

1: **if** X_1.type • X_2.type ∈ { XY, YX, XX, XYX, YXX, XXY } **then**
2: return create(join(X_1,X_2))
3: **else**
4: return create(join(backoff(X_1),X_2)) ∩ create(join(X_1,backoff(X_2)))
5: **end if**

The • operator in line 1 is the concatenation operator, used to check whether the merging of X_1 and X_2 creates a legal sequence of symbols, i.e. whether there is a rule that emits that sequence of symbols. If the merging creates a legal sequence, then the routine continues with the execution of the create() subroutine. Every time this routine is executed, the create() subroutine creates two items: one for the monotone reordering and one for the reverse reordering, setting the item's *op* variable accordingly. Otherwise, the routine merges one item with the back off version of the other item as specified in line 4. The backoff routine basically reverts the item's type to X.

The probability score of each newly created item can be calculated in a straightforward manner. For instance, the language model score can be directly

calculated as the target string can be constructed according to the concatenation operation specified by *op*. Likewise, the orientation model score can also be calculated since the item data type already stores the item's reordering operation. The calculation of the dominance model is also straightforward, since the information about the word alignment information is stored.

The calculation of the argument selection model requires more explanation. While the calculation of the grow model is straightforward as it can be calculated every time an argument is appended to a head, the calculation of the number of arguments and the stop model requires prior information about the full range of the item which is not known beforehand. In our implementation, we postpone the calculation of these two models up to the point where: the item is backed off (line 3) or the merging produces the maximum sequence of nonterminal symbols, i.e. the concatenation of X_1 and X_2 produces either XX, XYX, YXX, or XYY.

Appendix B

List of Function Words

We list down the 128 most frequent words used in experiments in Chapter 5 below. We mark the frequent words that are also function words with * symbol after the words.

,	的*	。	”	在*	一*	)	和*
(	是*	年	及*	日	了*	於	有
个	:	为*	会	中	对*	与*	不*
中国	说	将*	这	香港	上	以*	政府
人	他	发展	後	也*	新	;	我们
两	时	第	而*	月	由*	完	并*
三	就*	要*	至*	已*	大*	台湾	有关
问题	或*	经济	等*	署	工作	该	可*
都*	最*	关系	到*	美国	多*	进行	服务
所*	地*	项	计划	次	名	更*	被*
国家	表示	向*	下	我	能	来*	二
国	但*	内	提供	其*	亦*	元	国际
他们	处	各*	区	局	包括	社会	之*
合作	人士	者	从*	会议	高	前	今日

?	活动	每*	委员会	道	以及*	教育	市民
公司	著	电	人民	方面	四	重要	还*