Title	Tree-to-String Phrase-based Statistical Machine Translation
Author(s)	NGUYEN, Thai Phuong
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12070
Rights	
Description	Supervisor:島津　明, 情報科学研究科, 博士

Tree-to-String Phrase-based Statistical Machine Translation

by

NGUYEN PHUONG THAI

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Professor AKIRA SHIMAZU

School of Information Science
Japan Advanced Institute of Science and Technology

November 2007

To My Family

Abstract

The major aim of our study is to improve phrase-based statistical machine translation (SMT) using syntactic information represented in constituent tree form. In recent years, there have been many studies about syntactic SMT. Most studies rely on formal grammars such as synchronous context-free grammars and tree transducers. The approaches can be different in a number of aspects such as input type for example string or tree, in rule form for example SCFG or xRs, in rule function including word reordering or word choice. Since these studies aim to improve both word reordering and word choice, their grammars have been fully lexicalized. We would like to make a distinction between word order and word choice when statistically modelling the translation process. We suppose that the input of a SMT system is a syntactic tree. Considering word order as a syntactic problem, we define syntactic transformation task which involves the word reordering, the deletion and the insertion of function words. We propose a syntactic transformation model based on the probabilistic context free grammar. By using this model, we studied a number of tree-to-string phrase-based SMT approaches which vary in the way syntactic information is used including preprocessing and decoding and the level of syntactic analysis including chunking and parsing. Our experimental results showed significant improvements in translation quality. Considering word choice as a semantic problem, we aim at incorporating WSD into phrase-based SMT. Our empirical study on this problem reveal various aspect of the integration. Our experiments showed a significant improvement in translation quality.

Key words: Computational Linguistics, Statistical Machine Translation, Syntactic Parsing, Word Reordering, Word Sense Disambiguation, Word Choice.

Acknowledgments

In nearly three years of my PhD student life, there are many people help me stay focused and sane in pursuit of my goal.

First of all, I would like to express my deepest thanks to my advisor, Professor Akira Shimazu, for giving me the freedom to explore many paths of research and giving me guidance along the way. He also manages to ensure that my projects are always going somewhere useful. I gratefully appreciate his patient supervision, encouragement, and support over the years. I am proud to be one of his students.

I would like to say my special thanks to Professor Ho Tu Bao for his help during my PhD life. He gives me and other Vietnamese students lots of good advice not only for study but also for life. Besides, I have got valuable experiences from working with him.

I wish to say grateful thanks to Associate Professor Kiyoaki Shirai for gentle discussions. Besides, he always asks many questions in seminars of Vietnamese students. Since we often feel free when answering his questions, a seminar becomes more enjoyable.

I would like to thank Dr. Eiichiro Sumita and Associate Professor Kentaro Torisawa for giving comments on my thesis. Their comments help me improve both my thesis and presentation.

Besides, JAIST has been a great place for me to develop myself as a researcher. I receive a lot of inspiration, feedback, and insight that help me with my work, broaden my horizon from fellow students and researchers. I specially wish to say grateful thanks to Dr. Nguyen Le Minh, Lecturer Dam Hieu Chi, Assist. Prof. Huynh Van Nam, Assist. Prof. Makoto Nakamura, Mr. Kenji Takano, Dr. Le Anh Cuong, Mr. Nguyen Van Vinh, Mr. Nguyen Tri Thanh, Mr. Nguyen Anh Tuan, Dr. Phan Xuan Hieu, Dr. Doan Son, and Mr. Nguyen Canh Hao.

I would also like to express my grateful appreciation to my former teachers, Professor Ha Quang Thuy and Professor Dinh Manh Tuong, College of Technology, Vietnam National University, Hanoi (VNUH), for their kindly recommendations and constant encouragement before and during my research at JAIST. Without their helps, I could not receive the permission to go to JAIST. I also appreciate the help and the encouragement from Professor Ho Si Dam, Dr. Hoang Xuan Huan, Dr. Pham Hong Thai, and many other faculty members of College of Technology, VNUH.

I specially express my thanks and my respect to my former advisor, Dr. Pham Hong Nguyen. He has guided me to the way of research in Natural Language Processing. I am very proud to be one of his students, and also a member in his Machine Translation group.

Last, but not least, I am grateful to have a loving family who always care about me: my father Nguyen Van Ton, my mother Pham Thi Xuan, my younger sisters Nguyen Phuong Lan and Nguyen Phuong Thuy. I would not have survived these years without their love and friendship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we briefly state the research context, our motivations, as well as the major contributions of this thesis. Firstly, we briefly introduce the problem of machine translation and its important role in natural language processing. Secondly, we state the research problems which this thesis attempts to solve as well as the main motivations behind the work. Next, the main contributions of the thesis are shortly mentioned. Finally, the structure of the thesis will be outlined.

1.1 Overview

1.1.1 Machine Translation Problem

Natural language processing is one of the basic research fields of artificial intelligence. This research field studies how to create computer programs which can process human language. There are various problems concerning the human's language ability ranging from very fundamental tasks such as morphological analysis, syntactic parsing, and word sense disambiguation to application problems such as machine translation and text summarization. NLP becomes more and more important because of the rapid growth of text documents and the need for automated text processing. Solving NLP problems are a long term dream of human.

Human translation requires linguistic knowledge of both source and target languages such as morphology, syntactics, semantics, pragmatics, and so on. Those knowledge are necessary to resolve the ambiguities of natural languages which exist at various levels. Machine translation also wants the same ability.

As an application of NLP, machine translation can claim to be one of the oldest field of study. One of the first proposed non-numerical use of computers. Up to now there are still five main obstacles to machine translation. The first difficulty is a word choice problem. The fundamental task solving this problem is word sense disambiguation (WSD). The second difficulty is a word order problem. For example, English has a SVO sentence structure while Japanese SOV. The third is tense and aspect. It is difficult to translate a

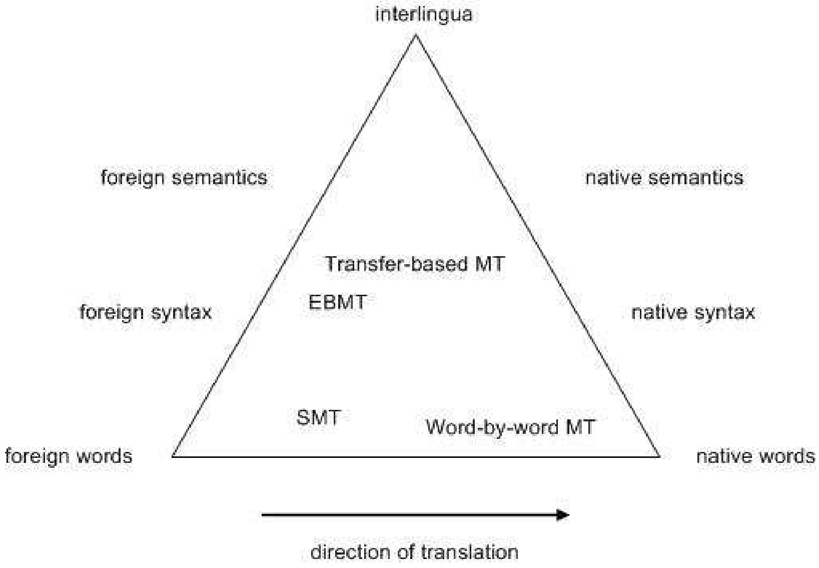


Figure 1.1: Levels of the use of linguistic knowledge according to various MT approaches

Vietnamese paragraph into English with a correct and coherent tense and aspect through the whole paragraph. The fourth difficulty is pronoun translation. This problem must be solved at the document level. The fifth obstacle is idiom translations. It is very difficult to collect all possible idioms.

Statistical machine translation is an approach to MT which is based on learning knowledge from bilingual corpora. Bitext contains parallel documents of two languages. Translation patterns are learned from bitext automatically. At the first days of SMT, patterns are word translation. Recently, by using phrase as the basic unit of translation [Koehn et al., 2003], SMT achieves a big step. Figure 1.1 illustrates levels of the use of linguistic knowledge according to various MT approaches. Conventional phrase-based SMT [Koehn et al., 2003] almost belongs to the lowest level. There have been a growing trend towards employing advances in NLP for SMT[1]. We review this trend according to involved NLP tasks including morphological analysis, POS tagging, syntactic parsing, and word sense disambiguation.

1.1.2 Morphological Analysis and POS Tagging

Morphological analysis was used to deal with the data sparseness problem [Goldwater & McClosky (2005)]. Words could be transformed using various ways in the SMT preprocessing phase. Many kinds of information were used such as word surface form, lemma, tense, case, etc. Koehn and Hoang (2007) proposed a factored translation model for phrase-based SMT. The authors modelled phrase-to-phrase translation as a generative process utilizing information at word level such as POS tag, lemma, case, etc. Using this kind of information is very useful for language pairs which are different in morphology.

[1] Another interesting trend is to applying new machine learning methods such as discriminative training to improve SMT.

Approach	Input	Theoretical model	Rule form
Koehn et al. (2003)	string	FSTs	no
Yamada and Knight (2001)	string	SCFGs	SCFG rule
Melamed (2003)	string	SCFGs	SCFG rule
Chiang (2005)	string	SCFGs	SCFG rule
Quirk et al. (2005)	dependency tree	Tree transducers	Treelet pair
Galley et al. (2006)	string	Tree transducers	xRs rule
Liu et al. (2006)	tree	Tree transducers	xRs rule
Our work	tree	SCFGs	SCFG rule

Table 1.1: A comparison of syntactic SMT approaches (part 1). FST=Finite State Transducer; SCFG=Synchronous Context-Free Grammar; xRs is a kind of rule which maps a syntactic pattern to a string, for example VP(AUX(does), RB(not),x_0:VB) → ne, x_0, pas.

Approach	Decoding style	Linguistic information	Phrase usage	Performance
Koehn et al. (2003)	beam search	no	yes	baseline
Yamada and Knight (2001)	parsing	target	no	not better
Melamed (2003)	parsing	both sides	no	not better
Chiang (2005)	parsing	no	yes	better
Quirk et al. (2005)	parsing	source	yes	better
Galley et al. (2006)	parsing	target	yes	better
Liu et al. (2006)	tree transformation	source	yes	better
Our work	tree transformation	source	yes	better

Table 1.2: A comparison of syntactic SMT approaches (part 2)

1.1.3 Word Sense Disambiguation

Aiming to improve word choice ability, [Varea et al. (2001)] studied context sensitive lexical models. However, contextual features used by this study were not as rich as state-of-the-art WSD models [Ando(2006)]. Most recently, several studies focused on integrating WSD into SMT. Those studies were motivated by an observation that SMT made decision based on local context through translation models and language models rather than the context at sentence level or even document level. WSD did not have such limitations. There were successful integrations of WSD into phrase-based SMT [Carpuat and Wu (2007)] and hierarchical phrase-based SMT [Chan et al. (2007)].

1.1.4 Syntactic Parsing

In the previous sub-section, we only mentioned SMT systems which were weighted finite-state transducers (WFSTs) of the ”phrase”-based variety, meaning that they memorize the translations of word n-grams, rather than just single words. To advance the state of the art, SMT system designers began to experiment with tree-structured translation models [Yamada and Knight (2001), Melamed (2004), Marcu et al. (2006)]. The underlying computational models were synchronous context-free grammars and weighted finite-state tree transducers which conceptually have a better expressive power than WFSTs.

We create Tables 1.1, 1.2, and 1.3 in order to compare syntactic SMT approaches

Approach	Rule function	Rule lexicalization level
Koehn et al. (2003)	no	no
Yamada and Knight (2001)	reorder and function-word ins./del.	unlexicalized
Melamed (2003)	reorder and word choice	full
Chiang (2005)	reorder and word choice	full
Quirk et al. (2005)	word choice	full
Galley et al. (2006)	reorder and word choice	full
Liu et al. (2006)	reorder and word choice	full
Our work	reorder and function-word ins./del.	half

Table 1.3: A comparison of syntactic SMT approaches (part 3). In the column Rule lexicalization level: full=lexicalization using vocabularies of both source language and target language; half=using source vocabulary and function words of target vocabulary.

Property	English	Vietnamese	French	Japanese
Word delimiter	Space	No	Space	No
Inflection	Suffixing	Using function words	Suffixing	Suffixing
Derivation	Suffixing	Using functionhi words	Suffixing	Suffixing
Sentence word order	SVO	SVO	SVO	SOV
Adjective modifier order	Preceding	Following	Both	Preceding
Determiner modifier order	Preceding	Both	Preceding	Preceding
Numeral modifier order	Preceding	Preceding	Preceding	Preceding
Possessor modifier order	Preceding	Following	Preceding	Preceding
Relative clause order	Following	Following	Following	Preceding
Ad-position	Preposition	Preposition	Preposition	Postposition
Interrogative word position	Wh-movement	Wh-in-situ	Wh-movement	Wh-in-situ
Topic prominent	No	Yes	No	Yes

Table 1.4: A comparison of several linguistic properties between English, Vietnamese, French, and Japanese. The properties in italic point out relative position of a modifier to its head noun. Languages without wh-movement are referred to as wh-in-situ languages.

including ours. The first row is a baseline phrasal SMT approach. The second column in Table 1.1 only describes input types because the output type is always string. Syntactic SMT approaches are different in many aspects. Most approaches which make use of phrases (in either explicit or implicit way) can beat the baseline approach (Table 1.2). What can we infer from this observation? Researchers have used more complex patterns (than phrase), and with the support of machine learning methods, they have advanced the state of the art. Two main problems these models aim to deal with is word order and word choice. In order to accomplish this purpose, the underlying formal grammars (including synchronous context-free grammars and tree transducers) are fully lexicalized (Table 1.3).

1.2 Motivations

The conventional phrase-based statistical machine translation (SMT) approach makes use of linguistic knowledge little or indirectly (Och and Ney, 2004), while there are many available high-performance linguistic tools such as parser, POS tagger or named entity recognizer (Manning and Schutze, 2003). An ideal phrase-based SMT system should take advantage of both bilingual corpora and linguistic analysis tools. For example, since phrases are limited in length, the word-order difference between languages is an obstacle for phrase-based SMT. Using morphological and syntactic information is a systematic way to deal with this problem. Our experiments involve three language pairs: English-Vietnamese, English-French, and English-Japanese. By surveying literatures [Kuno (1981), Gunji (1987), Cook (1988), Dung (2003)], we create Table 1.4 comparing a number of linguistic properties of these four languages.

In the field of compiler, the syntax-directed translation schemata has a dominant influence. A compiler is likely to carry out several or all of the following operations: lexical analysis, preprocessing, parsing, semantic analysis, code generation, and code optimization. After the parsing step, the syntactic structure of a program is identified. The parse tree is often analyzed, augmented, and transformed by later phases in the compiler. Those phases are controlled by syntax. A similar schema is used for natural language translation in transfer-based approaches. Since natural languages are highly ambiguous, various techniques and resources have been employed for dealing with ambiguity. For example, parsing natural languages requires chart parsing algorithms such as CYK and Earley which can deal with CFG languages, while compilers use faster parsing algorithms for sub-classes of CFG languages. We also want a kind of syntax-control ability for SMT.

Word order can be considered as a syntactic problem. Conversely, word choice is a semantic problem. Recently, in parsing research topic, [Klein and Manning (2003), Bikel (2004), Petrov et al. (2006)] have shown that un-lexicalized or lightly lexicalized parsers can achieve very high parsing accuracy. On the contrary, in WSD research topic, [Lee & Ng(2002)] have shown that lexical context made of words and POS tags has a main contribution to the performance of WSD systems. These observations suggest us to bring the discrimination between word order and word choice into designing a SMT system. We will design a reordering model which does not involve word choice for SMT. We also use WSD for SMT.

We aim to improve translation quality in which phrase-based SMT is considered as the baseline approach. We deal with two major problems: word order and word choice. Generally, for each problem, we design and use new feature functions. Each function is a probabilistic distribution and it takes into account a new kind of knowledge. For word order problem, we design a syntactic transformation model. This model requires syntactic knowledge of the source language. For word choice problem, we use WSD models. These models make use of non-local context including sentence level or paragraph level.

1.3 Our Approach

Figure 1.2 is an illustration of the conceptual architecture of our work. In a translation process, there are two major tasks: word choice and surface form generation. The first task is mainly concerned with the translation model, the language model, and the WSD model. The second task involves the syntactic transformation model at both local and non-local levels. The language model and the translation model can only impact this task locally. There are *weak links* between word choice and syntactic transformation and

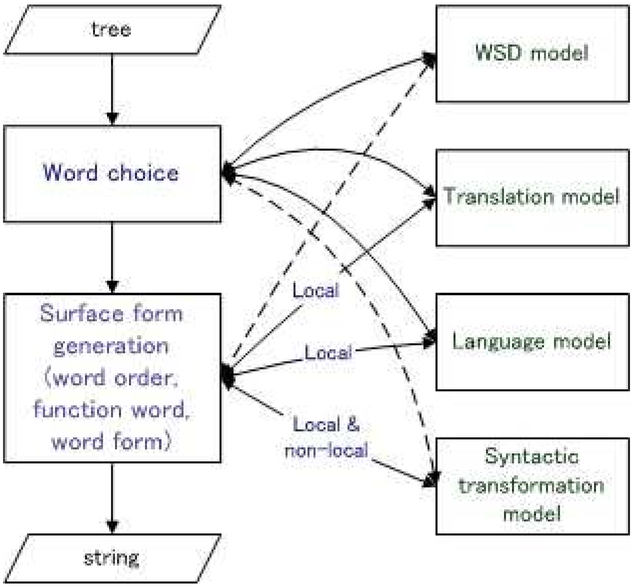

tree
Word choice
Surface form generation (word order, function word, word form)
string
WSD model
Translation model
Language model
Syntactic transformation model
Local
Local
Local & non-local

Figure 1.2: Conceptual architecture of our work.

between surface form generation and WSD. In fact, the distinctions between the two tasks are relative.

We study a number of tree-to-string phrase-based SMT approaches demonstrated in Figures 1.3 and 1.4. The name ”tree-to-string” simply means the input of a SMT system is syntactic trees[2]. Suppose that an input sentence has been parsed resulting in a phrase-structure tree of the source language. This tree will be reordered and translated into a string of the target language. The strength of phrases is not given up since the unit of translation is still phrases. Those approaches vary in syntactic analysis level (shallow or deep) and in how syntax is used for a SMT system (preprocessing or decoding). The postprocessing phase concerns with re-ranking n-best lists of possible target sentences, while our approach makes use of source linguistic information. Therefore we do not consider this phase. The properties of our approach are summarized in Tables 1.1 and 1.2. A number of advantages are as follow:

- Since translation is separated from parsing, parsers of the source language can be exploited.

- Since morphosyntactic information of the source side is made use of, more control over the translation process can be taken. In Figure 1.2, we have shown the relation between syntactic transformation and the translation process. We consider surface form generation is controlled (directed) by syntax. For example, about word order problem, without syntactic information (represented as a constituent tree), finding the best possible target word order is a NP-hard problem [Knight (1999)].

- Do not require syntactic information of the target side since for many languages good parsers are still not available.

- The rule form is simpler than other syntactic SMT approaches'.

[2] This name does not follow the noisy channel's regulation.

- Do not give up the strength of the baseline approach: phrases.
- Can achieve better performances than the baseline approach.

Following this approach, we have to deal with a number of issues:

- Design a syntactic transformation model
- Study different kinds of tree-to-string phrase-based (T2S PB) SMT:

  T2S PB SMT which uses syntactic transformation in the preprocessing phase: This approach can be used to build a T2S PB SMT system from scratch or by using an existing phrase-based SMT system (black box).

  T2S PB SMT which uses shallow-syntax transformation in the decoding phase: This kind of SMT system has an advantage of fast decoding.

  T2S PB SMT which employs full syntactic structure in the decoding phase (a general framework).
- Integrate WSD into PB SMT (can apply to all T2S PB approaches)

In the next section, we will describe how we deal with these issues and describe them in details.

1.4 Main Contributions

We defined syntactic transformation including the word reordering, the deletion and the insertion of function words. This definition prevents our model from learning heavy grammars to solve the word choice problem. We proposed a syntactic transformation model based on the lexicalized probabilistic context-free grammar [Thai & Shimazu (2006a)]. Since this model is sensitive with both structural and lexical information, it can deal with transformational ambiguity. It is trained by using a bilingual corpus, a word alignment tool, and a broad coverage parser of the source language. The parser is a constituency analyzer which can produce parse tree in Penn Tree-bank's style. The model is applicable to language pairs in which the target language is poor in resources.

We studied a phrase-based SMT approach [Thai & Shimazu (2006b)] which uses linguistic analysis in the preprocessing phase. The linguistic analysis includes morphological transformation and syntactic transformation. Since the word-order problem is solved using syntactic transformation, there is no reordering in the decoding phase. For morphological transformation, we used hand-crafted transformational rules. Specifically, we present a morphological transformation schema for English-Vietnamese translation. Our various experiments, which were carried out with several language pairs such as English-Vietnamese and English-French, showed significant improvements in translation quality.

A number of MT applications such as Web translation require high speed. Since full parsing may be slow for such applications, we consider chunking as an alternative. We study a chunking-based reordering method for phrase-based SMT [Thai & Shimazu (2007)]. We employ the syntactic transformation model for phrase reordering within chunks. The transformation probability is also used for scoring translation hypotheses.

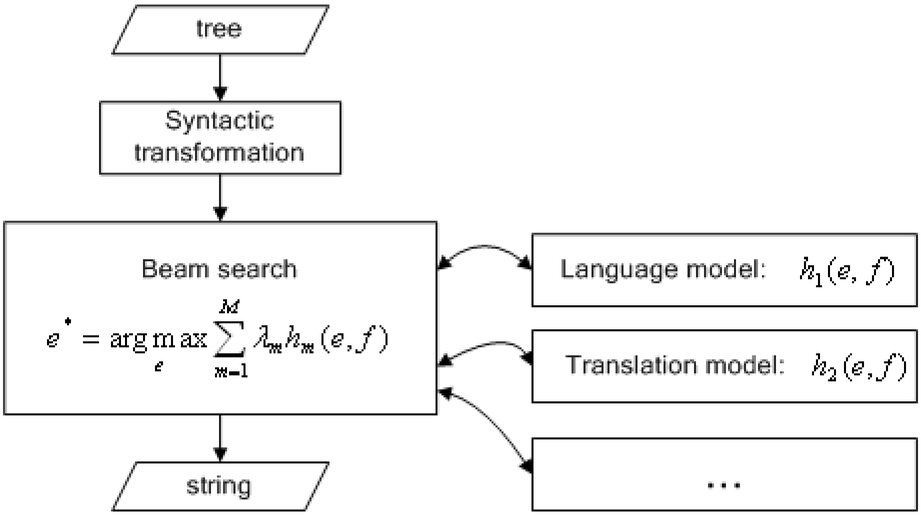

tree
Syntactic transformation
Beam search
$e^* = \arg\max_e \sum_{m=1}^{M} \lambda_m h_m(e,f)$
Language model: $h_1(e,f)$
Translation model: $h_2(e,f)$
...
string

Figure 1.3: SMT with syntactic transformation in the preprocessing phase.

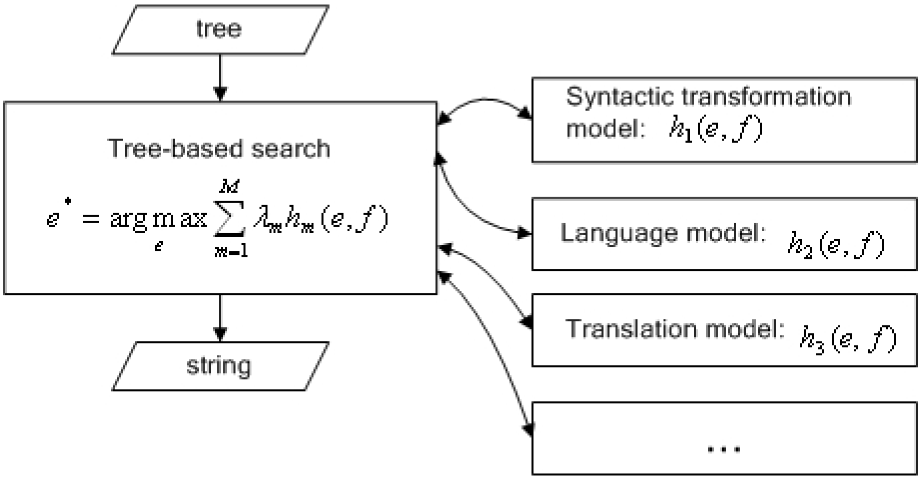

tree
Tree-based search
$e^* = \arg\max_e \sum_{m=1}^{M} \lambda_m h_m(e,f)$
Syntactic transformation model: $h_1(e,f)$
Language model: $h_2(e,f)$
Translation model: $h_3(e,f)$
...
string

Figure 1.4: SMT with syntactic transformation in the decoding phase.

Chunk reordering is carried out in the decoding phase. This study shows another way to apply the syntactic transformation model to SMT.

Two tree-to-string SMT approaches have been mentioned, one with preprocessing and the other with decoding but limited to shallow syntactic structures. In order to overcome this limitation, we consider another phrase-based SMT approach based on stochastic syntax-directed translation schemata. We propose a tree transformation algorithm and a tree-based decoding algorithm. The transformation algorithm converts a tree with word leaves into a tree with phrase leaves (phrase tree). The decoding algorithm is a dynamic programming algorithm which processes an input tree in a bottom-up manner. The syntactic transformation model is employed to control and score reordering operations. The chunking-based translation approach can be considered as an instance of this approach. We conducted experiments with English-Vietnamese and English-Japanese language pairs. Experimental results showed a significant improvement in terms of translation quality.

Beside the word order problem, word choice is another obstacle for MT. Though phrase-based SMT has an advantage of word choice based on local context, exploiting larger context is an interesting research topic. We carried out an empirical study of integrating WSD into SMT. We implemented the approach proposed by [Carpuat and Wu (2007)]. Our experiments reinformed that WSD can improve SMT significantly. We used two WSD models including MEM and NB while [Carpuat and Wu (2007)] used an ensemble of four combined WSD models (NB, MEM, Boosting, and Kernel PCA-based) and [Chan et al. (2007)] employed SVM. We evaluated WSD accuracy, effect of phrase length, the use of syntactic relation feature for SMT.

We built a SMT system for phrase-based log-linear translation models. This system has three decoders: beam search, chunking-based, and syntax-based. We used the system for our experiments with reordering and WSD.

1.5 Thesis Structure

The dissertation can be summarized in the eight main chapters as follows.

- The first chapter presents the overall view of the thesis including an introduction of statistical machine translation, motivations, our approach and contributions.
- In the second chapter, we review previous works in SMT.
- In the third chapter, we present our syntactic transformation model.
- In the fourth chapter, we present empirical results of phrase-based SMT with morphosyntactic transformation in the preprocessing phase.
- In the fifth chapter, we present chunking-based reordering for SMT.
- In the sixth chapter, we present a stochastic syntax-directed phrase-based SMT approach.
- In the seventh chapter, we present our empirical results of the integration of WSD into SMT.
- Finally, in the eighth chapter, we draw several conclusions from our works.

Chapter 2

Related Works

2.1 Phrase-Based SMT

The noisy channel model is the basic model for phrase-based SMT [Koehn et al. (2003)]:

$$\arg\max_{e} P(e|f) = \arg\max_{e}[P(f|e) \times P(e)] \tag{2.1}$$

The model can be described as a generative story [1]. First, an English sentence e is generated with probability $P(e)$. Second, e is segmented into phrases $\overline{e}_1, ..., \overline{e}_I$ (assuming a uniform probability distribution over all possible segmentations). Third, e is reordered according to a distortion model. Finally, French phrases $\overline{f}_i$ are generated under a translation model $P(\overline{f}|\overline{e})$ estimated from the bilingual corpus. Though other phrase-based models follow a joint distribution model [Marcu and Wong (2002)], or use log-linear models [Och and Ney (2004)], the basic architecture of phrase segmentation, phrase reordering, and phrase translation remains the same.

As discussed in [Och (2003)], the direct translation model represents the probability of English target sentence $e = \overline{e}_1, ..., \overline{e}_I$ being the translation for a French source sentence $f = \overline{f}_1, ..., \overline{f}_J$ through an exponential, or log-linear model:

$$p_\lambda(e|f) = \frac{exp(\sum_{k=1}^{m} \lambda_k \times h_k(e, f))}{\sum_{e' \subset E} exp(\sum_{k=1}^{n} \lambda_m \times h_k(e, f))} \tag{2.2}$$

where e is a single candidate translation for f from the set of all English translations E, λ is the parameter vector for the model, and each h_k is a feature function of e and f.

[1] We follow the convention in [Brown et al. (1993)], designating the source language as "French" and the target language as "English".

2.2 Weighted Synchronous Context-Free Grammars

[Chiang (2005)] proposed a hierarchical phrase-based SMT model which was formally a weighted synchronous CFG (Aho and Ullman, 1969). The rule form is:

$$X \rightarrow < \gamma, \alpha, \sim > \tag{2.3}$$

where X is a nonterminal, γ and α are string of both terminals and nonterminals, and $\sim$ is a one-to-one mapping between nonterminal occurrences in γ and in α. The following sentence pair is annotated with square brackets representing a possible hierarchical phrase structure:

[Aozhou] [shi] [[[yu [Bei Han] you [bangjiao]] de [shaoshu goujia]] zhiyi]

[Australia] [is] [one of [the [few countries] that [have [dipl. rels.] with [North Korea]]]]

The rules can be:

$$X \rightarrow \text{<yu } X_1 \text{ you } X_2\text{, have } X_2 \text{ with } X_1\text{>}$$

$$X \rightarrow \text{<}X_1 \text{ de } X_2\text{, the } X_2 \text{ that } X_1\text{>}$$

$$X \rightarrow \text{<}X_1 \text{ zhiyi, one of } X_1\text{>}$$

where $\sim$ is represented by indices.

The author used only one nonterminal symbol instead of assigning syntactic categories to phrases. Two special rules were added to combine sequence of Xs to form an S (the starting symbol):

$$S \rightarrow < S_1 X_2, S_1 X_2 > \tag{2.4}$$

$$S \rightarrow < X_1, X_1 > \tag{2.5}$$

This kind of grammar can capture word order and word choice well. It can be learned from bitext without any syntactic annotation. Since phrases are encoded directly in CFG rules (seen as hierarchical phrases), and there are only two non-terminal symbols, the grammar relies very much on lexical knowledge. Chiang's decoder was a CKY parser with beam search. His SMT system achieved better performance than the Pharaoh system [Koehn et al. (2003)] on a Chinese-English translation task.

[Melamed (2004)] used synchronous context-free grammars (SCFGs) for parsing both languages simultaneously. Melamed's study showed that syntax-based SMT systems could be built using synchronous parsers. He also discussed binarization of multi-text grammars on a theoretical level, showing the importance and difficulty of binarization for efficient synchronous parsing.

2.3 Dependency Treelet Translation

[Quirk et al. (2005)] proposed a translation model which incorporates dependency representation of the source and target languages. The authors supposed that input sentences were parsed by a dependency parser. An advantage in comparison with the phrase-based approach is that the dependency structure can capture non-local dependency between words such as *ne...pas*(*not*). Since it is difficult to specify reordering information within elementary dependency structures, the authors used a separate reordering model. This reordering model is sensitive with lexical information such as words and POS tags. [Quirk et al. (2005)] reported better BLEU scores than the Pharaoh system on an English-French translation task.

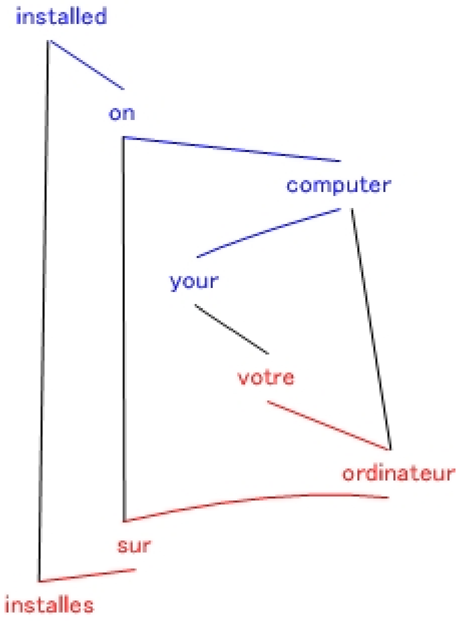


Figure 2.1: An English-French treelet translation pair.

[Quirk et al. (2005)] defined a treelet is an arbitrary connected subgraph of a dependency tree. The unit of translation is treelet pairs (Figure 2.1). Treelet translation pairs can be learned from bitext in which the source text has been parsed by a dependency parser. Given a word aligned sentence pair and a source dependency tree, the authors used the alignment to project the source structure onto the target sentence. Their decoding algorithm is influenced by ITG (Wu, 1997). They used a log-linear translation model (Och and Ney, 2002).

2.4 Tree-to-String Noisy Channel

The first approach can be called statistical machine translation by parsing. [Yamada and Knight (2001)] proposed a SMT model that uses syntax information in the target language alone. The model is based on a tree-to-string noisy channel model, and the translation task is transformed into a parsing problem.

(Galley et al., 2006) proposed a translation model based on weighted tree-to-string (xRs) transducers (Graehl and Knight, 2004). Their transformational rules r_i are equivalent to 1-state xRs transducers mapping a given pattern to a string. For example, *"does not"* can be transform into *"ne...pas"* in French by using the following rule:

$$\text{VP(AUX(does), RB(not),}x_0\text{:VB)} \rightarrow \text{ne, } x_0\text{, pas}$$

The left hand side of r_i can be arbitrary syntax tree fragment. Its leaves are either lexicalized or variables. The right hand side of r_i is represented as a sequence of target language words and variables. This kind of rule can capture context-rich syntax of the target language. The authors trained their model using the EM algorithm. Their decoding algorithm was based on tree transformation. Their experimental results were higher than those of (Galley et al., 2004). According to their point of view, since the input sentence is fixed and is generally already grammatical, it is less benefit in modelling the source language syntax.

2.5 Tree-to-String Alignment Template

This study [Liu et al. (2006)] is based on tree-to-string transducers [Graehl and Knight (2004)] but the source language syntax is modelled. Rules are learned from bitext in which the source text has been parsed. Their system, Lynx, achieved a performance higher than Pharaoh. Under their experimental settings, the number of rules was only one forth the number of bilingual phrases. The system can gain further improvement if both bilingual phrases and rules are used.

In order to enhance the expressive power of their model, [Liu et al. (2007)] proposed forest-to-string rule. A rule is a map from a sequence of subtree to a string. This kind of rule can cover non-syntactic phrase pairs better than tree-to-string rule. The new rule form leads to an improvement in translation quality over their original model.

[Liu et al. (2007)] discussed about how the phenomenon of non-syntactic bilingual phrases is dealt with in other SMT approaches. [Galley et al. (2004)] handled non-constituent phrasal translation by traversing the tree upwards until reaches a node that subsumes the phrase. [Marcu et al. (2006)] reported that approximately 28% of bilingual phrases are non-syntactic on their English-Chinese corpus. They proposed using a pseudo nonterminal symbol that subsumes the phrase and corresponding multi-headed syntactic structure. One new xRs rule is required to explain how the new nonterminal symbol can be combined with others. This technique brought a significant improvement in performance to their string-to-tree noisy channel SMT system.

2.6 Preprocessing and Postprocessing

A simple approach to the use of syntactic knowledge is to focus on the preprocessing phase. [Xia and McCord (2004)] proposed a preprocessing method to deal with the word-order problem. During the training of a SMT system, rewrite patterns were learned from bitext by employing a source language parser and a target language parser. Then at testing time, the patterns were used to reorder the source sentences in order to make their word order similar to that of the target language. The method achieved improvements over a baseline French-English SMT system. [Collins et al. (2005)] proposed reordering rules for restructuring German clauses. The rules were applied in the preprocessing phase of a

German-English phrase-based SMT system. Their experiments showed that this method could also improve translation quality significantly.

Reranking [Shen et al. (2004), Och et al. (2004)] is a frequently-used postprocessing technique in SMT. However, most of the improvement in translation quality has come from the reranking of non-syntactic features, while the syntactic features have produced very small gains [Och et al. (2004)].

2.7 Syntax-Based Language Model

[Charniak et al. (2003)] proposed an alternative approach to using syntactic information for SMT. The method employs an existing statistical parsing model as a language model within a SMT system. Experimental results showed improvements in accuracy over a baseline syntax-based SMT system.

2.8 Integration of WSD into SMT

Conventional phrase based systems use local context information from phrase table and language model. Though phrase based SMT achieves a jump in translation quality in comparison with word based SMT, there are still cases in which local context can not capture well the correct meaning of source words. WSD can use features from much larger contexts and those features can overlap each other. The idea of integrating WSD and SMT rises naturally from this perspective.

Varea et al. (2001) directly used context sensitive lexical models for SMT. Their SMT system was a word-based MEM. They reported significant decreases in perplexities of training and testing corpora. Besides, they also used these lexical models for re-ranking n-best lists and achieved slight improvements in translation quality.

Carpuat and Wu (2005) described their first effort to directly use a state-of-the-art WSD system for SMT. They used a word-based translation model, the IBM Model 4. All trials did not achieve any significant improvement in translation quality. They used WSD in three phases of SMT including preprocessing, decoding, and postprocessing. This empirical study seemed casting the doubt that: does WSD improve SMT? But unimproved assumption.

Cabezas and Resnik (2005) reported their positive results though not statistical significant when they applied WSD techniques to support a phrase-based SMT system. A WSD model was used to create a context sensitive word translation model. This model was trained using data generated from bilingual corpus. Words in target language are considered as senses. Then this word translation model was integrated into the SMT system since the baseline SMT system allows integration of alternative translation models. Carpuat et al. (2006) had the same approach as Cabezas and Resnik (2005) when they joined the IWSLT 2006. More than WSD, they also used NER to strengthen the semantic processing ability of a phrase-based SMT system. Those studies have same limitations that using WSD for single words and WSD have not integrated into SMT as a feature.

[Chan et al. (2007)] made use of WSD for hierarchical phrase-based translation. WSD training data was generated from bilingual corpus using word alignment information.

They used two new WSD features for SMT and proposed an algorithm for scoring synchronous rules. Phrases which does not exceed a length of two were computed WSD models. Their experiments, carried out using a standard Chinese to English translation task, showed that WSD can improve SMT significantly.

Simultaneously with [Chan et al. (2007)], [Carpuat and Wu (2007)] used a similar approach to the problem. The main difference was that they focused on conventional phrase-based SMT [Koehn et al. (2003)] and used only one WSD feature for SMT. The limit of phrase length was the same as the value used by their SMT system. Their experiments led to the same conclusion: WSD can improve SMT.

2.9 MT Evaluation Methods

2.9.1 BLEU

BLEU[2] [Papineni et al., 2002] is currently one of the most popular metric in the field. The central idea behind the metric is that, "the closer a machine translation is to a professional human translation, the better it is". The metric calculates scores for individual segments, generally sentences, and then averages these scores over the whole corpus in order to reach a final score. It has been shown to correlate highly with human judgements of quality at the corpus level. The quality of translation is indicated as a number between 0 and 1 and is measured as statistical closeness to a given set of good quality human reference translations. Therefore, it does not directly take into account translation intelligibility or grammatical correctness. BLEU should be used in a restricted manner, for comparing the results from two similar systems, and for tracking "broad, incremental changes to a single system" [Callison-Burch et al. (2006)]. BLEU score can be computed as:

$$Score(e,r) = BP(e,r) \times exp(\frac{1}{N} \times \sum_{n=1}^{N} log(p_n)) \tag{2.6}$$

where p_n represent the precision of n-grams suggested in e and BP is a brevity penalty measuring the relative shortness of e over the whole corpus.

2.9.2 WER

Word Error Rate (WER) is computed as the minimum number of substitution, deletion, and insertion operations that have to performed to convert a MT output sentence to a reference sentence. This metric is very sensitive to word order. Word error rate can be calculated as:

$$WER = \frac{S+D+I}{N} \tag{2.7}$$

[2] BiLingual Evaluation Understudy

where S is the number of substitutions, D is the number of the deletions, I is the number of the insertions, and N is the number of words in the reference.

2.9.3 PER

A shortcoming of the WER is the fact that it requires a perfect word order. The word order of an acceptable sentence can be different from that of the target sentence, so that the WER measure alone could be misleading. To overcome this problem, the position-independent word error rate (PER) has been proposed. This measure compares the words in the two sentences ignoring the word order.

Chapter 3

A Syntactic Transformation Model

For syntactic transformation, we propose a transformational model [Nguyen and Shimazu (2006a)] based on the probabilistic context free grammar (PCFG). The model's knowledge is learned from bitext in which the source text has been parsed.[1] The model can be applied to language pairs, in which the target language is poor in resources in the sense that it lacks of syntactically-annotated corpora and good syntactic parsers.

3.1 Motivations and Assumptions

When designing this model, we have a number of motivations:

- it is used for transforming syntactic structures of the source language into those of the target language
- under several assumptions stated later, syntactic transformation deals with the word order problem, the deletion and the insertion of function words. Syntactic transformation does not solve the word choice problem which is primarily concerned with word sense disambiguation.

We make several assumptions as follows:

- The source language is generated by a PCFG $G_s = (N, \Sigma, R_s, S)$ where
  - N = a finite set of nonterminals including constituent tags and part-of-speech (POS) tags;
  - $\Sigma = \Sigma_f \cup \Sigma_c$ where Σ_f is a finite set of function words and Σ_c is a finite set of content words of the source language;
  - R_s = a finite set of rules of the form $p : A \rightarrow \alpha$ for A in N, α in $N^* \cup \Sigma$, $0 < p \leq 1$;
  - S in N = the starting symbol.

[1] By a statistical parser trained on the Penn Wall Street Journal treebank [Marcus et al. (1993)].

- The target language is generated by a PCFG $G_t = (N, \Delta, R_t, S)$ where
  - $\Delta = \Delta_f \cup \Sigma_c$ where Δ_f is a finite set of function words of the target language;
  - R_t = a finite set of rules of the form $p : B \rightarrow \beta$ for B in N, β in $N^* \cup \Delta$, $0 < p \leq 1$;
- From the previous assumptions: G_s and G_t are only different in the set of function words and the rule set; R_s and R_t have two rule forms, one with a sequence of nonterminals on the right hand side (RHS), the other with a word on the RHS. These rule forms are compatible with CFG rules directly extracted from Penn Treebank.
- A basic transformation operation is a conversion of a rule $A \rightarrow \alpha$ in G_s into a rule $A \rightarrow \beta$ in G_t. The nonterminals in β, which are constituent label or are POS label of content words, are a permutation of those in α.

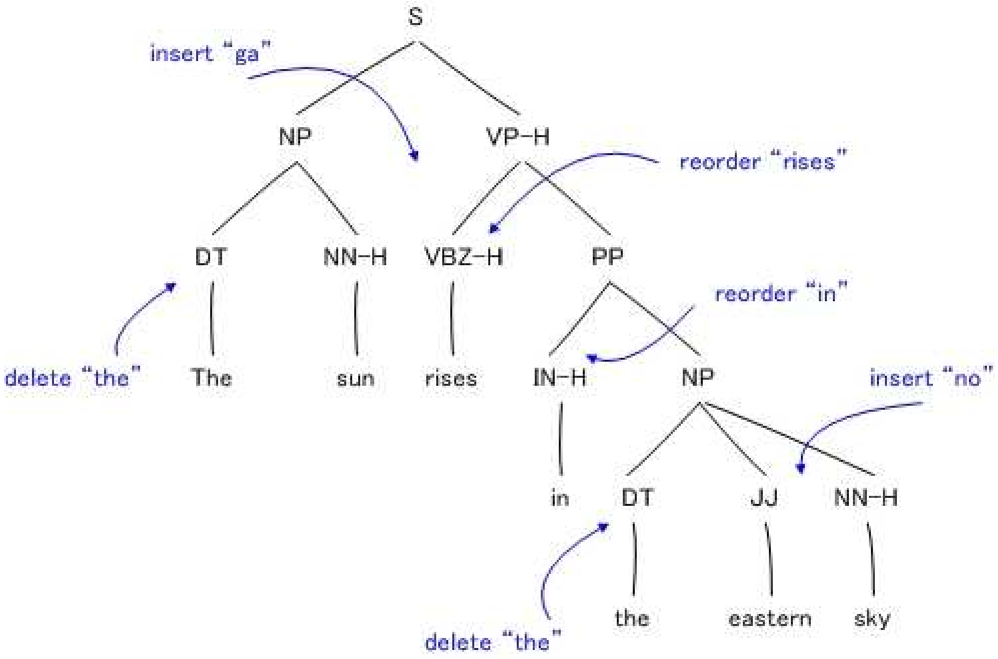


Figure 3.1: An English syntactic tree with possible transformations into a plausible Japanese syntactic structure.

We consider an example of English-Japanese syntactic transformation as follows:

English sentence: *"The sun rises in the eastern sky."*

Japanese sentence: *"taiyo ga | higashi no | sora ni | noboru"*

"sun SUBJECT | east POSSESSIVE | sky LOCATIVE | rise"

The syntactic tree is shown in Figure 3.1. The transformed tree is in Figure 3.2. These two trees can be generated by two simple grammars described later. These grammars satisfy the assumptions.

A simple English grammar (rule probability is omitted):

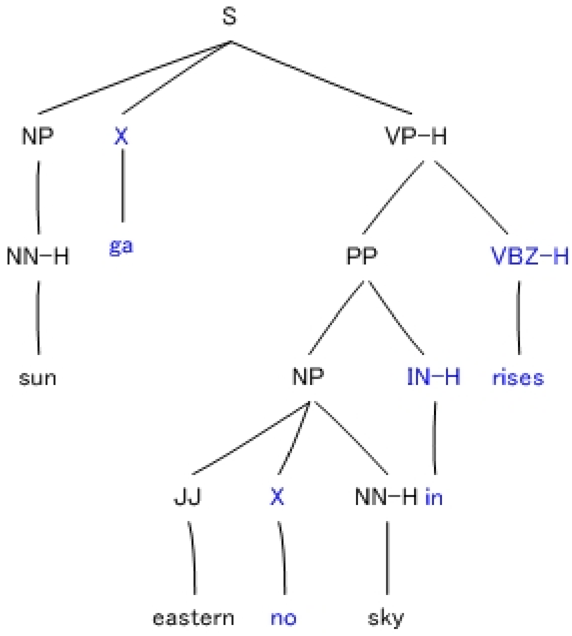


Figure 3.2: An English syntactic tree after transformed into a Japanese syntactic structure.

$N = \{S,\ NP,\ VP,\ PP,\ DT,\ NN,\ VBZ,\ IN,\ JJ,\ X\}$

$\Sigma_f = \{the,\ in\}$

$\Sigma_c = \{sun,\ rises,\ eastern,\ sky\}$

$R_s = \{S \rightarrow NP\ VP\text{-}H;\ NP \rightarrow DT\ NN\text{-}H;\ VP \rightarrow VBZ\text{-}H\ PP;\ PP \rightarrow IN\text{-}H\ NP;$

$NP \rightarrow DT\ JJ\ NN\text{-}H;\ DT \rightarrow the;\ NN \rightarrow sun \mid sky;\ VBZ \rightarrow rises;\ IN \rightarrow in;$

$JJ \rightarrow eastern\}$

S is the start symbol.

A possible corresponding Japanese grammar:

$N = \{S,\ NP,\ VP,\ PP,\ DT,\ NN,\ VBZ,\ IN,\ JJ,\ X\}$

$\Delta_f = \{ga,\ no,\ in\}$

$\Delta_c = \Sigma_c = \{sun,\ rises,\ eastern,\ sky\}$

$R_t = \{S \rightarrow NP\ VP\text{-}H;\ NP \rightarrow NN\text{-}H;\ VP \rightarrow PP\ VBZ\text{-}H;\ PP \rightarrow NP\ IN\text{-}H;$

$NP \rightarrow JJ\ X\ NN\text{-}H;\ NN \rightarrow sun \mid sky;\ VBZ \rightarrow rises;\ IN \rightarrow in;\ X \rightarrow ga \mid no;$

$JJ \rightarrow eastern\}$

S is the start symbol.

3.2 Syntactic Transformation Model

One major difficulty in the syntactic transformation task is ambiguity. There can be many different ways to reorder a CFG rule. For example, the rule[2] $NP \rightarrow DTJJNN$ in English can become $NP \rightarrow DTNNJJ$ or $NP \rightarrow NNJJDT$ in Vietnamese. For the phrase "a nice girl", the first reordering is most appropriate, while for the phrase "this weekly radio", the second one is correct. Lexicalization of CFG rules is one way to deal with this problem. Therefore we propose a transformational model which is based on probabilistic decisions and also exploits lexical information.

3.2.1 Transformational Model

Suppose that S is a given lexicalized tree of the source language (whose nodes are augmented to include a word and a part of speech (POS) label). S contains n applications of lexicalized CFG rules $LHS_i \rightarrow RHS_i$, $1 \leq i \leq n$, (LHS stands for left-hand-side and RHS stands for right-hand-side). We want to transform S into the target language word order by applying transformational rules to the CFG rules. A transformational rule is represented as a pair of unlexicalized CFG rules *TR=(LHS → RHS, LHS → RHS')*. For example, the rule *(NP → JJ NN, NP → NN JJ)* implies that the CFG rule *NP → JJ NN* in source language can be transformed into the rule $NP \rightarrow NN\ JJ$ in target language. Since the possible transformational rule for each CFG rule is not unique, there can be many transformed trees. The problem is how to choose the best one. Suppose that T is a possible transformed tree whose CFG rules are annotated as $LHS_i \rightarrow RHS'_i$, which is the result of converting $LHS_i \rightarrow RHS_i$ using a transformational rule TR_i. Using the Bayes formula, we have:

$$P(T|S) = \frac{P(S|T) \times P(T)}{P(S)} \tag{3.1}$$

The transformed tree T^* which maximizes the probability $P(T|S)$ will be chosen. Since $P(S)$ is the same for every T, and T is created by applying a sequence Q of n transformational rules to S, we can write:

$$Q^* = \arg\max_Q [P(S|T) \times P(T)] \tag{3.2}$$

The probability $P(S|T)$ can be decomposed into:

$$P(S|T) = \prod_{i=1}^{n} P(LHS_i \rightarrow RHS_i | LHS_i \rightarrow RHS'_i) \tag{3.3}$$

where the conditional probability $P(LHS_i \rightarrow RHS_i | LHS_i \rightarrow RHS'_i)$ is computed with

[2] NP: noun phrase, DT: determiner, JJ: adjective, NN: noun

the unlexicalized form of the CFG rules. Moreover, we constraint:

$$\sum_{RHS_i} P(LHS_i \rightarrow RHS_i | LHS_i \rightarrow RHS'_i) = 1 \tag{3.4}$$

To compute $P(T)$, a lexicalized probabilistic context free grammar (LPCFG) can be used. LPCFGs are sensitive to both structural and lexical information. Under a LPCFG, the probability of T is:

$$P(T) = \prod_{i=1}^{n} P(LHS_i \rightarrow RHS'_i) \tag{3.5}$$

Since application of a transformational rule only reorders the right-hand-side symbols of a CFG rule, we can rewrite (3.2):

$$\begin{aligned} &Q^* = \{TR^*_i : TR^*_i = \arg\max_{TR_i}[P(LHS_i \rightarrow RHS_i | LHS_i \rightarrow RHS'_i) \times \\ &P(LHS_i \rightarrow RHS'_i)], i = 1, .., n\} \end{aligned} \tag{3.6}$$

3.2.2 Markovization of Lexicalized CFG Rules

Suppose that a lexicalized CFG rule has the following form:

$$F(h) \rightarrow L_m(l_m)...L_1(l_1)H(h)R_1(r_1)...R_k(r_k) \tag{3.7}$$

where $F(h)$, $H(h)$, $R_i(r_i)$, and $L_i(l_i)$ are all lexicalized non-terminal symbols; $F(h)$ is the left-hand-side symbol or parent symbol, h is the pair of head word and its POS label; H is a head child symbol; and $R_i(r_i)$ and $L_i(l_i)$ are right and left modifiers of H. Either k or m may be 0, k and m are 0 in unary rules. Since the number of possible lexicalized rules is huge, direct estimation of $P(LHS \rightarrow RHS')$ is not feasible. Fortunately, some LPCFG models [Collins (1999), Charniak (2000)] can compute the lexicalized rule's probability efficiently by using the rule-markovization technique [Collins (1999), Charniak (2000), Klein and Manning (2003)]. Given the left hand side, the generation process of the right hand side can be decomposed into three steps:

1. Generate the head constituent label with probability $P_H = P(H|F, h)$

2. Generate the right modifiers with probability $P_R = \prod_{i=1}^{k+1} P(R_i(r_i)|F, h, H)$ where $R_{k+1}(r_{k+1})$ is a special symbol which is added to the set of nonterminal symbols. The grammar model stops generating right modifiers when this symbol is generated (STOP symbol.)

3. Generate the left modifiers with probability $P_L = \prod_{i=1}^{m+1} P(L_i(l_i)|F, h, H)$ where $L_{m+1}(l_{m+1})$ is the STOP symbol.

This is zeroth order markovization (the generation of a modifier does not depend on previous generations). Higher orders can be used if necessary. The probability of a lexicalized CFG rule now becomes $P(LHS \rightarrow RHS') = P_H \times P_R \times P_L$.

The LPCFG which we used in our experiments is Collins' Grammar Model 1 [Collins (1999)]. We implemented this grammar model with some linguistically-motivated refinements for non-recursive noun phrases, coordination, and punctuation [Collins (1999), Bikel (2004)]. We trained this grammar model on a treebank whose syntactic trees resulted from transforming source language trees. In the next section, we will show how we induced this kind of data.

3.3 Training

The required resources and tools include a bilingual corpus, a broad-coverage statistical parser of the source language, and a word alignment program such as GIZA++ [Och and Ney (2000)]. First, the source text is parsed by the statistical parser. Then the source text and the target text are aligned in both directions using GIZA++. Next, for each sentence pair, source syntactic constituents and target phrases (which are sequences of target words) are aligned. From this hierarchical alignment information, transformational rules and a transformed syntactic tree are induced. Then the probabilities of transformational rules are computed. Finally, the transformed syntactic trees are used to train the LPCFG. We can summarize the description in the following steps:

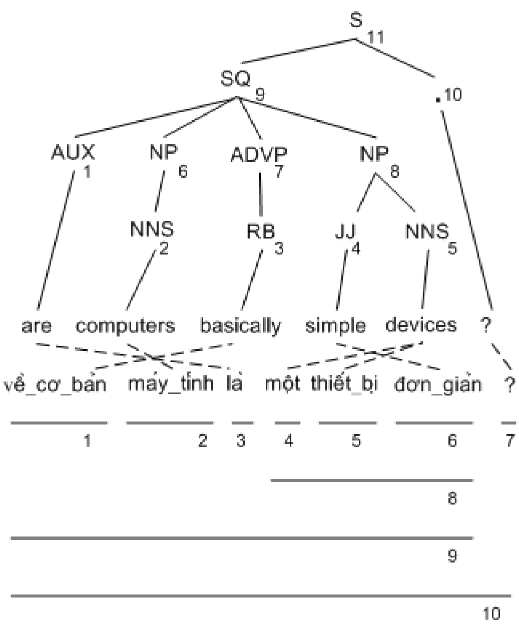
- Step 1: Word alignment, parsing

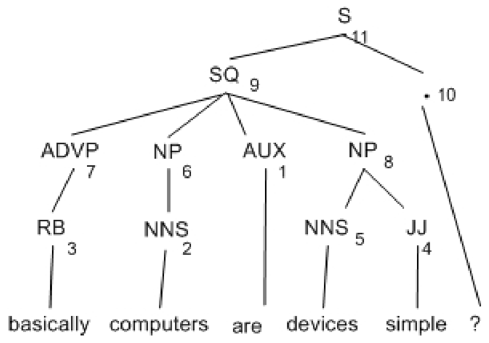
- Step 2: Hierarchical alignment (project source sentence structure on target sentence)

- Step 3: Rule induction

- Step 4: Parameter estimation

Figure 3.3 shows an example of inducing transformational rules for English-Vietnamese translation. Source sentence and target sentence are in the middle of the figure, on the left. The source syntactic tree is at the upper left of the figure. The source constituents are numbered. Word links are represented by dotted lines. Words and aligned phrases of the target sentence are represented by lines (in the lower left of the figure) and are also numbered. Word alignment results, hierarchical alignment results, and induced transformational rules are in the lower right part of the figure. The transformed tree is at the upper right.


S 11
SQ 9
. 10
AUX 1
NP 6
ADVP 7
NP 8
NNS 2
RB 3
JJ 4
NNS 5
are computers basically simple devices ?
về_cơ_bản máy_tính là một thiết_bị đơn_giản ?
1 2 3 4 5 6 7
8
9
10



Word alignment: 1-3, 2-2, 3-1, 4-6, 5-4, 5-5, 10-7

Hierarchical alignment: 6-2, 7-1, 8-8, 9-9, 11-10

Transformational rules: (S -> SQ ., 0 1), (SQ -> AUX NP ADVP NP, 2 1 0 3), (NP -> NNS, 0), (ADVP -> RB, 0), (NP -> JJ NNS, 1 0)

Figure 3.3: Inducing transformational rules

3.3.1 Hierarchical Alignment

To determine the alignment of a source constituent, link scores between its span and all of the target phrases are computed using the following formula [Xia and McCord (2004)]:

$$score(s,t) = \frac{links(s,t)}{words(s) + words(t)} \tag{3.8}$$

where s is a source phrase, t is a target phrase; links(s,t) is the total number of source words in s and target words in t that are aligned together; words(s) and words(t) are, respectively, the number of words in s and t. A threshold is used to filter bad alignment possibilities. After the link scores have been calculated, the target phrase, with the highest link score, and which does not conflict with the chosen phrases will be selected. Two target phrases do not conflict if they are separate or if they contain each other.

We supposed that there are only one-to-one links between source constituents and target phrases. We used a number of heuristics to deal with ambiguity. For source constituents whose span contains only one word which is aligned to many target words, we choose the best link based on the intersection of directional alignments and on word link score. When applying formula (3.8) in determining alignment of a source constituent, if there were several target phrases having the highest link score, we used an additional criterion:

- for every word outside s, there is no link to any word of t
- for every word outside t, there is no link to any word of s

3.3.2 Target CFG Rule Induction

Given a hierarchical alignment, transformational rules can be computed for each constituent of the source syntactic tree. Suppose that X is a source constituent with children $X_0, ..., X_n$. $Y_0, ..., Y_m$ is a sequence of target phrases in which Y_j are sorted increasingly according to the index of their first word. The criteria for inducing a transformational rule are as follows:

- Y_j are adjacent to each other.
- for each X_i, it is aligned to a target phrase Y_j or its span is a non-aligned word.
- for each Y_j, it is aligned to a source constituent X_i or it is a non-aligned word.

If those criteria are satisfied, a transformational rule can be induced:
$(X \rightarrow X_0...X_n, X \rightarrow Z_0...Z_m)$ where $Z_j = X_i$ if Y_j is aligned to X_i or $Z_j = Y_j$ if Y_j is a non aligned word.

For example, in Fig. 4.1, the constituent SQ (9) has four children AUX_0 (1), NP_1 (6), $ADVP_2$ (7), and NP_3 (8). Their aligned target phrases are[3]: Y (9), Y_0 (1), Y_1 (2), Y_2 (3), and Y_3 (8). The target-source alignment is

1-7 (Y_0-$ADVP_2$)

2-6 (Y_1-NP_1)

3-1 (Y_2-AUX_0)

8-8 (Y_3-NP_3)

Since all criteria are satisfied, a transformation rule is induced:
$(SQ \rightarrow AUXNPADVPNP, SQ \rightarrow ADVP_2NP_1AUX_0NP_3)$

3.3.3 Insertion and Deletion

The example in the previous sub-section does not have insertion or deletion operations. Now we consider another example demonstrated in Figure 3.4. In this figure we use another representation style (different from Figure 3.3) showing the hierarchical alignment more clearly. The 1-0 word alignments trigger deletion operations (*the*). The 0-1 word alignments trigger insertion operations (*ga* and *no*). *ga* and *no* are assigned default label X. Figure 3.5 shows a part of step 3 in which two transformation rules $(NP \rightarrow DTNN, NP \rightarrow NN)$ and $(NP \rightarrow DTJJNN, NP \rightarrow JJXNN)$ are induced. Since $X \rightarrow no$ is not aligned to any source word, we use an heuristic which allocates that node to the first node subsuming it (NP in this case).

[3] For clarity, we use Y symbol instead of Vietnamese phrases.

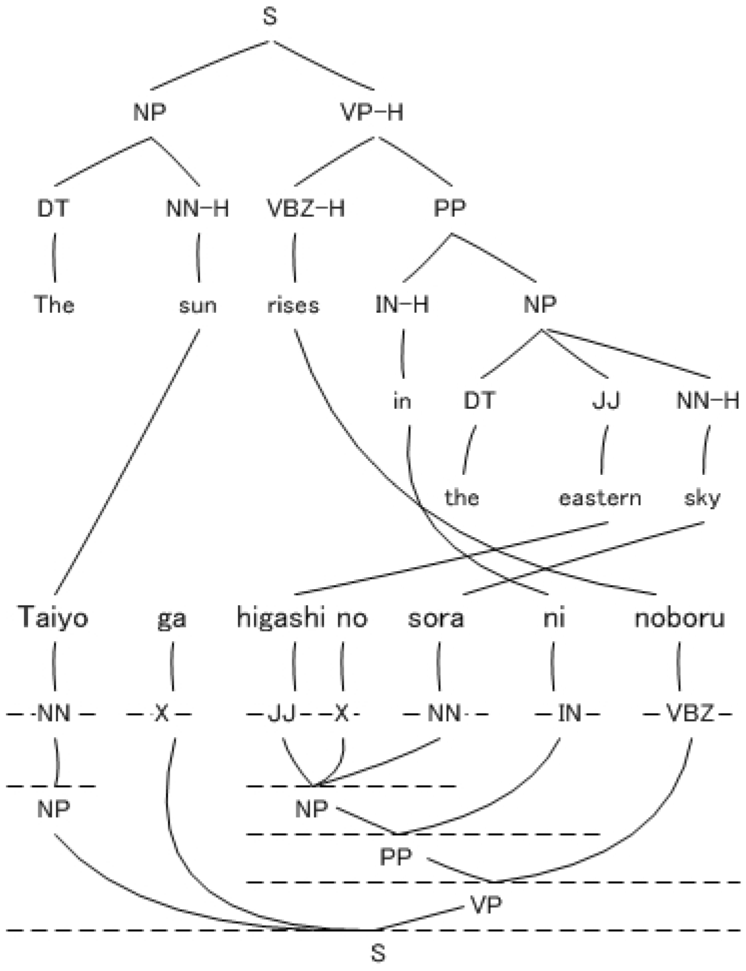


Figure 3.4: Transformation rule induction for English-Japanese translation: step 1&2.

3.3.4 Transformed Trees

For a sentence pair, after transformational rules have been induced, the source syntactic tree will be transformed. The constituents which do not have a transformational rule remain unchanged (all constituents of the source syntactic tree in Fig. 3.3 have a transformational rule). Their corresponding CFG rule applications are marked as untransformed and are not used in training the LPCFG.

3.3.5 Parameter Estimation

The conditional probability for a pair of rules is computed using the maximum likelihood estimate:

$$P(LHS \rightarrow RHS|LHS \rightarrow RHS') = \frac{Count(LHS \rightarrow RHS, LHS \rightarrow RHS')}{Count(LHS \rightarrow RHS')} \qquad (3.9)$$

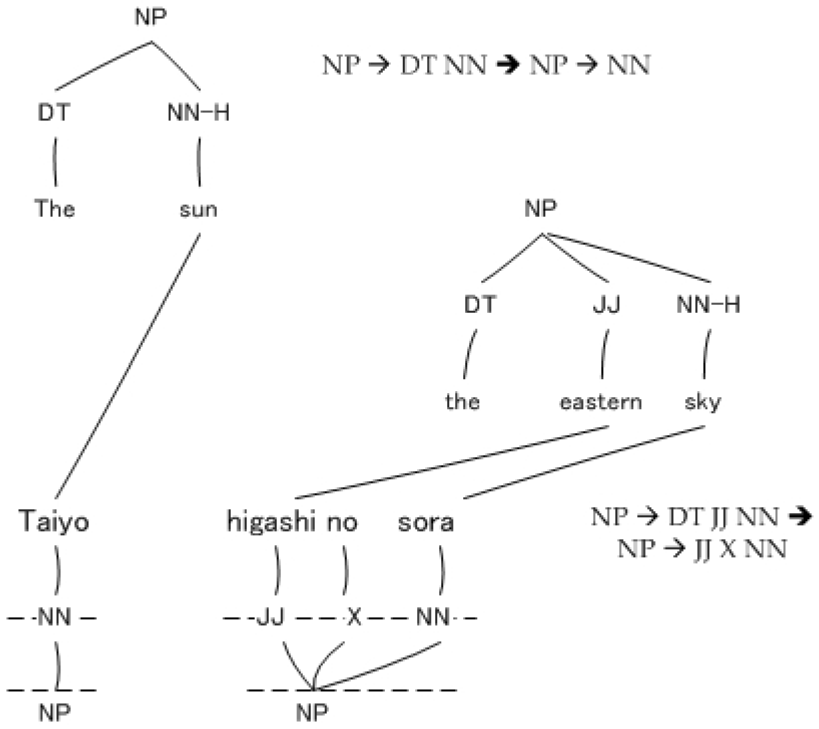


Figure 3.5: Transformation rule induction for English-Japanese translation: part of step 3.

In training the LPCFG, a larger number of parameter classes have to be estimated such as head parameter class, modifying nonterminal parameter class, and modifying terminal parameter class. Very useful details for implementing Collins' Grammar Model 1 were described in [Bikel (2004)].

3.4 Applying

After it has been trained, the transformational model is used in the preprocessing or decoding phase of an SMT system. Given a source syntactic tree, first the tree is lexicalized by associating each non-terminal node with a word and a part of speech (computed bottom-up, through head child).[4] Next, the best sequence of transformational rules is computed by formula (3.6). Finally, by applying transformational rules to the source tree, the best transformed tree is generated.

3.5 Conclusion

For syntactic transformation, we have proposed a transformational model based on the probabilistic context free grammar and a technique for inducing transformational rules

[4] For a CFG rule, which symbol in the right hand side is the head was determined using heuristic rules [Collins (1999)].

from source-parsed bitext. Our method can be applied to language pairs in which the target language is poor in resources. In the future, we would like to extend the transformational model to deal with non-local transformations. Moreover, we intend to use EM algorithm to estimate the transformation probability better.

Chapter 4

Improving Phrase-Based SMT with Morpho-Syntactic Transformation

We present a phrase-based SMT approach which uses linguistic analysis in the preprocessing phase. The linguistic analysis includes morphological transformation and syntactic transformation. Since the word-order problem is solved using syntactic transformation, there is no reordering in the decoding phase. For morphological transformation, we use hand-crafted transformational rules. For syntactic transformation, we employ the transformational model described in the previous chapter. This phrase-based SMT approach is applicable to language pairs in which the target language is poor in resources. We considered translation from English to Vietnamese and from English to French. Our experiments showed significant BLEU-score improvements in comparison with Pharaoh, a state-of-the-art phrase-based SMT system.

4.1 Introduction

In the field of statistical machine translation (SMT), several phrase-based SMT models [Och and Ney (2004), Marcu and Wong (2002), Koehn et al. (2003)] have achieved state-of-the-art performance. These models have a number of advantages in comparison with the original IBM SMT models [Brown et al. (1993)] such as word choice, idiomatic expression recognition, and local restructuring. These advantages are the result of moving from words to phrases as the basic unit of translation.

Although phrase-based SMT systems have been successful, they have some potential limitations when it comes to modelling word-order differences between languages. The reason is that the phrase-based systems make little or only indirect use of syntactic information. In other words, they are still ”non-linguistic”. That is, in phrase-based systems tokens are treated as words, phrases can be any sequence of tokens (and are not necessarily phrases in any syntactic sense), and reordering models are based solely on movement distance [Och and Ney (2004), Koehn et al. (2003)] but not on the phrase content. Another

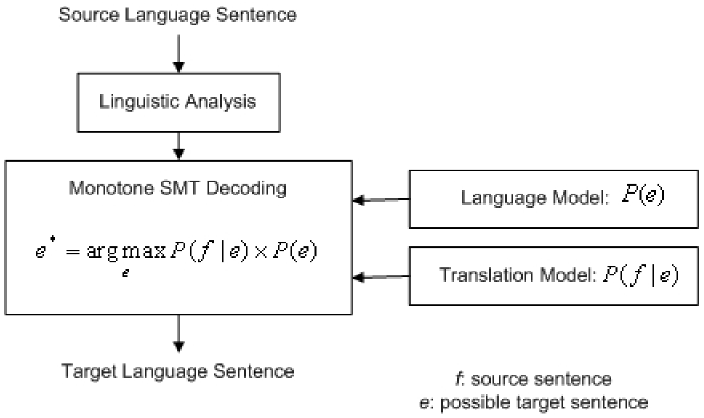


Figure 4.1: The architecture of our SMT system

+ Step 1:	Parse the source sentence
+ Step 2:	Transform the syntactic tree
+ Step 3:	Analyze the words at leaf nodes morphologically to lemmas and suffixes
+ Step 4:	Apply morphological transformation rules
+ Step 5:	Extract the surface string

Table 4.1: Preprocessing procedure

limitation is the sparse data problem, because acquiring bitext is difficult and expensive. Since in phrase-based SMT differently inflected forms of the same word are often treated as different words, the problem is more serious when one or both of the source and target languages is an inflectional language.

In this chapter, we describe our study for improving SMT by using linguistic analysis in the preprocessing phase to attack these two problems (Figure 4.1). The word order problem is solved by parsing source sentences, and then transforming them into the target language structure. After this step, the resulting source sentences have the word order of the target language. In the decoding phase, the decoder searches for the best target sentence without reordering source phrases. The sparse data problem is solved by splitting the stem and the inflectional suffix of a word during translation. The preprocessing procedure which carries out morpho-syntactic transformations takes in a source sentence and performs five steps as shown in Table 4.1. This preprocessing procedure was applied to source sentences in both the training and testing phases.

In our experiments, we mainly considered translation from English to Vietnamese. Since there are significant differences between English and Vietnamese, this language pair is appropriate to demonstrate the effectiveness of the proposed method. We used Pharaoh [Koehn (2004)] as a baseline phrase-based SMT system. Our experiments showed significant improvements of BLEU score. We analyzed these experiments with respect to morphological transformation, syntactic transformation, and their combination. However, since our English-Vietnamese corpora are small, we carried out other experiments for the English-French language pair on the Europarl corpus [Koehn et al. (2003)], a large one. Within the range of data size with which we experimented, we found out that improvements made by syntactic transformation over Pharaoh do not change significantly as the corpus scales up.[1] Moreover, a SMT system with syntactic transformation needs shorter maximum phrase length to achieve the same translation quality as the baseline system.

The rest of this chapter is organized as follows: In Section 2, background information is presented. Section 3 presents the morphological transformation. Finally, Section 4 discusses our experimental results.

4.2 Background

4.2.1 Syntactic Preprocessing for SMT

Our study differs from those of [Xia and McCord (2004)] and [Collins et al. (2005)] in several important respects. First, our transformational model is based on the PCFG, while neither of the previous studies used probability in their reordering method. Second, the transformational model is trained by using bitext and only a source language parser, while [Xia and McCord (2004)] employed parsers of both source and target languages. Third, we use syntactic transformation in combination with morphological transformation. Last, we consider translation from English to Vietnamese and from English to French.

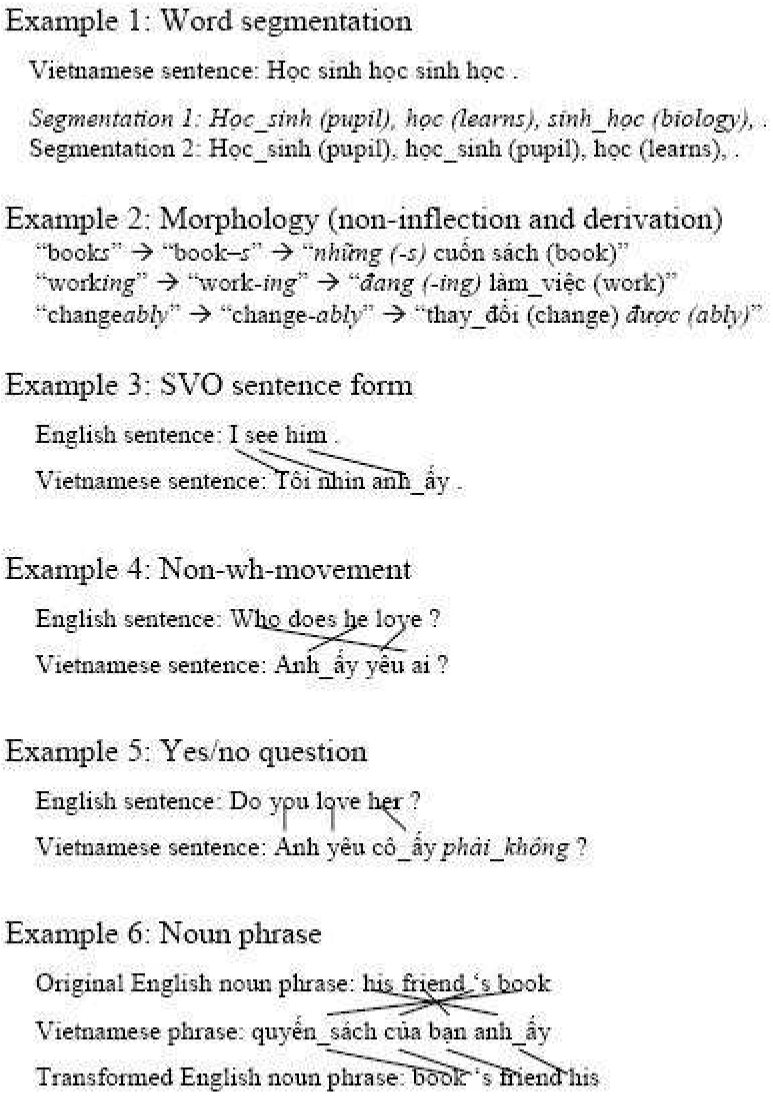
4.2.2 Morphological Analysis for SMT

According to our observations, most research on this topic has focused on preprocessing. [Al-Onaizan et al. (1999)] reported a study of Czech-English SMT which showed that improvements could be gained by utilizing morphological information. Some Czech processing tools, such as a morphological analyzer, part-of-speech (POS) tagger, and lemmatizer, are required. Ordinary Czech text can be changed in several different ways, including word lemmatization, attachment of morphological tags to lemmas, or the use of pseudo words. Each technique can be used separately or in combination with others to preprocess source texts before training or testing. [Niessen & Ney (2004)] presented a bag of useful techniques using morphological and shallowly syntactic information to improve German-English SMT. These techniques include: separating German verb prefixes, splitting German compound words, annotating some frequent function words with POS tags, merging phrases, and treating unseen words using their less specific forms. Another study of exploiting morphological analysis for Arabic-English SMT was reported in [Lee (2004)]. The method requires the morphological analysis of the Arabic text into morphemes, and the POS tagging of the bitext. After aligning the Arabic morphemes to English words, the system determines whether to keep each affix as a separate item, merge it back to the stem, or delete it. The choice of an appropriate operation relies on the consistency of the English POS tags that the Arabic morphemes are aligned to. [Goldwater & McClosky (2005)] recently used the techniques proposed in [Al-Onaizan et al. (1999)] for the Czech-English language pair, with some refinements, and analyzed the usefulness of each morphological

[1] Works which use morphological transformation for SMT have a property of vanishing improvement [Goldwater & McClosky (2005)].

feature. They also proposed a new word-to-word alignment model to use with the modified lemmas. Experimental results showed that the most significant improvement was achieved by combining the modified lemmas with the pseudo words.

4.2.3 Vietnamese Language Features



Example 1: Word segmentation

Vietnamese sentence: Học sinh học sinh học .

Segmentation 1: Học_sinh (pupil), học (learns), sinh_học (biology), .

Segmentation 2: Học_sinh (pupil), học_sinh (pupil), học (learns), .

Example 2: Morphology (non-inflection and derivation)

"books" → "book–*s*" → "*những (-s)* cuốn sách (book)"

"work*ing*" → "work-*ing*" → "*đang (-ing)* làm_việc (work)"

"change*ably*" → "change-*ably*" → "thay_đổi (change) *được (ably)*"

Example 3: SVO sentence form

English sentence: I see him .

Vietnamese sentence: Tôi nhìn anh_ấy .

Example 4: Non-wh-movement

English sentence: Who does he love ?

Vietnamese sentence: Anh_ấy yêu ai ?

Example 5: Yes/no question

English sentence: Do you love her ?

Vietnamese sentence: Anh yêu cô_ấy *phải_không* ?

Example 6: Noun phrase

Original English noun phrase: his friend 's book

Vietnamese phrase: quyển_sách của bạn anh_ấy

Transformed English noun phrase: book 's friend his

Figure 4.2: Examples

This section describes some phenomena specific to the Vietnamese language. The first is word segmentation. Like a number of other Asian languages such as Chinese, Japanese and Thai, Vietnamese has no word delimiter. The smallest unit in the construction of Vietnamese words is the syllable. A Vietnamese word can be a single word (one syllable) or a compound word (more than one syllable). A space is a syllable delimiter but not a word delimiter in Vietnamese. A Vietnamese sentence can often be segmented in many ways.

Feature	Describe	Vietnamese word order	Example
+pl	Plural noun	+pl noun	+pl book
+sg3	Third-person, singular, present-tense verb	+sg3 verb	+sg3 love
+ed	Past tense verb	+ed verb	+ed love
+ing	Present participle verb	+ing verb	+ing love
+pp	Past participle verb	+pp verb	+pp love
+er	Comparative adjective/adverb	adj/adv +er	small +er
+est	Superlative adjective/adverb	adj/adv +est	small +est

Table 4.2: Morphological features

Example 1 in Figure 4.2[2] shows a Vietnamese sentence with two possible segmentations. Obviously, Vietnamese word segmentation is a non-trivial problem.

The second phenomenon is morphology. Vietnamese is a non-inflectional language. Most English inflected word forms can be translated into a Vietnamese phrase. First, the word form is analyzed morphologically to a lemma and an inflectional suffix. Then the lemma is translated into a Vietnamese word which is the head of the phrase, and the suffix into a Vietnamese function word which precedes/follows and modifies the head word. English derivative words often correspond to Vietnamese compound words. Example 2 in Figure 4.2 shows a number of English words and their translations.

The third difference is word order. Vietnamese has a SVO sentence form[3] similar to English (see Example 3 in Figure 4.2.) However, wh-movement is significantly different between Vietnamese and English. In English, a wh-question starts with an interrogative word, while in Vietnamese, the interrogative word is not moved to the beginning of a wh-question (see Example 4 in Figure 4.2.) In addition, most Vietnamese yes/no questions end with an interrogative word, while the English yes/no questions do not (see Example 5 in Figure 4.2.) In phrase composition, the Vietnamese word order is quite different from English. The main difference is that in order to make an English phrase similar in word order to Vietnamese, we often have to move its pre-modifiers to follow the head word (see Example 6 in Figure 4.2.)

4.3 Morphological Transformation

In this research, we restricted morphological analysis to the inflectional phenomenon.[4] English inflected words were analyzed morphologically into a lemma and an inflectional suffix. Deeper analysis (such as segmenting a derivative word into prefixes, stem, and suffixes) was not used. We experimented with two techniques [Al-Onaizan et al. (1999)]: The first lemmatizes English words (lemma transformation[5]). The second one treats inflectional suffixes as pseudo words (which normally correspond to Vietnamese function words) and reorders them into Vietnamese word order if necessary (pseudo-word transformation). For example:

[2] For clarity, in the following sections, we use the underscore character '_' to connect the syllables of Vietnamese compound words.

[3] S stands for subject, V stands for verb, and O stands for object.

[4] This is due to the morphological analyzer that we used (section 5.1).

[5] In the rest of the chapter, the terms lemma or lemma transformation are used interchangeably.

Table 4.3: Corpora and data sets.

Corpus	Sentence pairs	Training set	Dev test set	Test set
Computer	8,718	8,118	251	349
Conversation	16,809	15,734	403	672
Europarl	740,000	95,924	2,000	1,122

Table 4.4: Corpus statistics of English-Vietnamese translation task.

		English	Vietnamese
Computer	Sentences	8,718	
	Average sentence length	20	21.5
	Words	173,442	187,138
	Vocabulary	8,829	7,145
Conversation	Sentences	16,809	
	Average sentence length	8.5	8
	Words	143,373	130,043
	Vocabulary	9,314	9,557

Source sentence: He has travelled to many famous places.
Lemmatized sentence: He have travel to many famous place.
Sentence with pseudo words: He +sg3 have +pp travel to many famous +pl place.

Our morphological features are listed fully in Table 4.2. In the next section, we will describe our experimental results in two cases: morphological transformation alone (lemma or pseudo-word), and in combination with syntactic transformation.

4.4 Experiments

4.4.1 Experimental Settings

We carried out experiments of translation from English to Vietnamese and from English to French. For the first language pair, we used two small corpora: one collected from some computer text books (named "Computer") and the other collected from some grammar books (named "Conversation"). For the second language pair, we used the freely available Europarl corpus [Koehn et al. (2003)]. Data sets are described in Tables 4.3, 7.1, and 4.5. For quick experimental turn around, we used only a part of the Europarl corpus for training. We created a test set by choosing sentences randomly from the common test part [Koehn et al. (2003)] of this corpus.

A number of tools were used in our experiments. Vietnamese sentences were segmented using a word-segmentation program [Nguyen et al. (2003)]. For learning phrase translations and decoding, we used Pharaoh [Koehn (2004)], a state-of-the-art phrase-based SMT system which is available for research purpose. For word alignment, we used the GIZA++ tool [Och and Ney (2000)]. For learning language models, we used SRILM

Table 4.5: Corpus statistics of English-French translation task.

		English	French
Training	Sentences	95,924	
	Average sentence length	27.8	32.4
	Words	2,668,158	3,109,276
	Vocabulary	29,481	39,661
Test	Sentences	1,122	
	Average sentence length	28	32
	Words	31,448	36,072
	Vocabulary	4,548	5,174

Table 4.6: Unlexicalized CFG rules (UCFGRs), transformational rule groups (TRGs), and ambiguous groups (AGs).

Corpus	UCFGRs	TRGs	AGs
Computer	4,779	3,702	951
Conversation	3,634	2,642	669
Europarl	14,462	10,738	3,706

toolkit [Stolcke (2002)]. For MT evaluation, we used BLEU measure [Papineni et al. (2001)] calculated by the NIST script version 11b. For the parsing task, we used Charniak's parser [Charniak (2000)] and another program [Johnson (2002)] for recovering empty nodes and their antecedents in syntactic trees. For morphological analysis, we used a rule-based morphological analyzer which is described in [Pham et al. (2003)].

4.4.2 Training the Transformational Model

On each corpus, the transformational model was trained resulting in a large number of transformational rules and an instance of Collins' Grammar Model 1. We restricted the maximum number of syntactic trees used for training the transformational model to 40000. Table 4.6 shows the statistics resulted from learning transformational rules. On three corpora, the number of transformational rule groups which were learned is smaller than the corresponding number of CFG rules. The reason is that there are many CFG rules which appear once or several times, however their hierarchical alignments did not satisfy the conditions of inducing a transformational rule. Another reason is that there were CFG rules which required nonlocal transformation.[6]

4.4.3 BLEU Scores

In each experiment, we ran Pharaoh's trainer with its default settings. Then we used Pharaoh's minimum-error-rate training script to tune feature weights to maximize the

[6] That is carried out by reordering subtrees, instead of reordering CFG rules. [Fox (2002)] investigated this phenomenon empirically for French-English language pair. [Knight and Graehl (2005)] presented a survey about tree automata which may be a useful way to deal with the non-local transformation.

Table 4.7: BLEU scores.

Corpus	Baseline	Lemma	PWord	Syntax	Lemma-Syntax	PWord-Syntax
Computer	47	46.88	48.5	50	50.03	51.94
Conversation	35.47	36.19	35.56	38.12	38.76	38.83
Europarl	26.41			28.02		

system's BLEU score on the development test set.

Experimental results on the test sets are shown in Table 4.7. The table shows the BLEU scores of the Pharaoh system (baseline) and other systems, which are formed by the combination of the Pharaoh system with various types of morpho-syntactic preprocessing. In column 2, the baseline score on the Computer corpus is higher than the baseline score on the Conversation corpus, due to differences between these corpora. Columns 3 and 4 show the scores in cases where the morphological transformation was used. Each of these scores is better than the corresponding baseline score. For the Computer corpus, the pseudo-word score is higher than the lemma score. Conversely, for the Conversation corpus, the pseudo-word score is not higher than the lemma score. Since the Computer corpus contains sentences (from computer books) in written language, the morphological features are translated quite closely into Vietnamese. In contrast, those features are translated more freely into Vietnamese in most of the Conversation corpus which contains spoken sentences. Therefore, the elimination of morphological features (by lemma transformation) in the Conversation corpus is less harmful than in the Computer corpus. Column 5 shows the BLEU scores when syntactic transformation is used. On each corpus, the syntax score is higher than the baseline score, and also higher than the score achieved by the system with morphological transformation. The last two columns, 6 and 7, show the scores of morpho-syntactic combinations. The combination of lemma and syntax is not very effective, because on both corpora, the lemma-syntax score is slightly higher than the score when using syntax alone, and the lemma-syntax improvement is no better than the total of individual improvements. However, on both corpora, the improvement made by combining pseudo word and syntax is better than the total of individual improvements.

Table 4.7 also contains experimental results on the Europarl corpus (columns 2 and 5).[7] The improvement made by syntactic transformation is only 1.61%. On the Vietnamese corpora, the corresponding improvements are 3% and 2.65%. The differences between those values can be explained in the following ways: First, we are considering the word order problem, so the improvement can be expected to be higher with language pairs which are more different in word order. According to our knowledge, Vietnamese and English are more different in word order than French and English. Second, by using phrases as the basic unit of translation, phrase-based SMT captures local reordering quite well if there is a large amount of training data.

[7] According to our knowledge, English and French are considered weakly inflected languages. Therefore we did not use morphological transformation for this language pair.

Table 4.8: Sign tests.

Test set	Subsets	Lemma	PWord	Syntax	Lemma-Syntax	PWord-Syntax	CValue
Computer	23 (15)	12/11	*17/6*	*20/3*	*18/5*	*21/2*	7
Conversation	22 (30)	14/8	12/10	*20/2*	*21/1*	*21/1*	6
Europarl	22 (51)			*17/5*			6

4.4.4 Significance Tests

In order to test the statistical significance of our results, we chose the sign test[8] [Lehmann (1986)]. We selected a significance level of 0.05. The Computer test set was divided into 23 subsets (15 sentences per subset), and the BLEU metric was computed on each of these subsets separately. The translation system with preprocessing was then compared to the baseline system over these subsets. For example, we found that the system with pseudo-word transformation had a higher score than the baseline system on 17 subsets, and the baseline system had a higher score on 6 subsets. With the chosen significance level of 0.05 and the number of subsets 23, the critical value is 7. So we can state that the improvement made by the system with pseudo-word transformation was statistically significant. The same experiments were carried out for the other systems (see Table 4.8.)

In columns 3 and 4, only the improvement gained by pseudo-word transformation on the Computer corpus is statistically significant. The other improvements, achieved by morphological transformation, are inconclusive. In contrast, all the improvements gained by syntactic transformation and morpho-syntactic combinations are statistically significant (columns 5, 6, and 7).

In addition to the tests reported so far, two other tests were carried out to verify the improvements of the pseudo word-syntax combination over syntax alone. The results were 18/5 on the Computer corpus and 16/6 on the Conversation corpus. These results mean that the improvements are significant. Therefore the combination is beneficial.

4.4.5 Some Analyses of the Performance of Syntactic Transformation

Figure 4.3 displays individual ngram precisions when syntactic transformation is used. Unigram precisions increase less than the others. These numbers confirm that the translation quality of long phrases increases and that syntactic transformation has a greater influence on word order than on word choice. Figure 4.4 contains some examples of better translations generated by the system using syntactic transformation.

A limitation of the syntactic transformation model is that it can not handle non-local transformation. By dealing with this kind of transformation, the BLEU score can be improved more. We give a linguistically-motivated example here. In English-Vietnamese translation, wh-movement belongs to non-local transformation (see Section 2.4). In a syntactic tree with a SBARQ symbol at the root, the wh-constituent (WHNP, WHADJP, WHPP, or WHADVP) is always co-indexed with a null element. Therefore the tree is

[8] Sign test was also used in [Collins et al. (2005)].

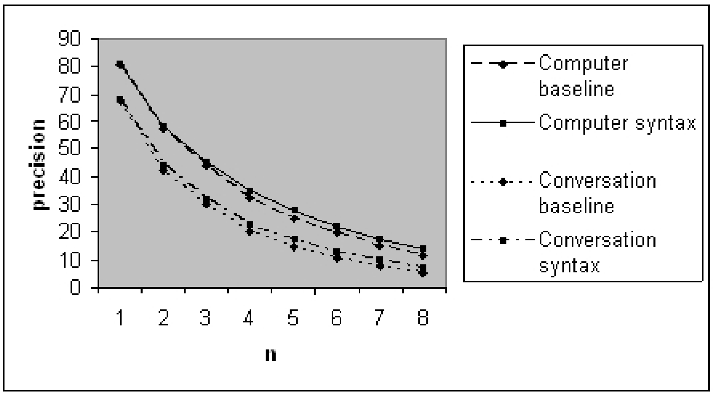
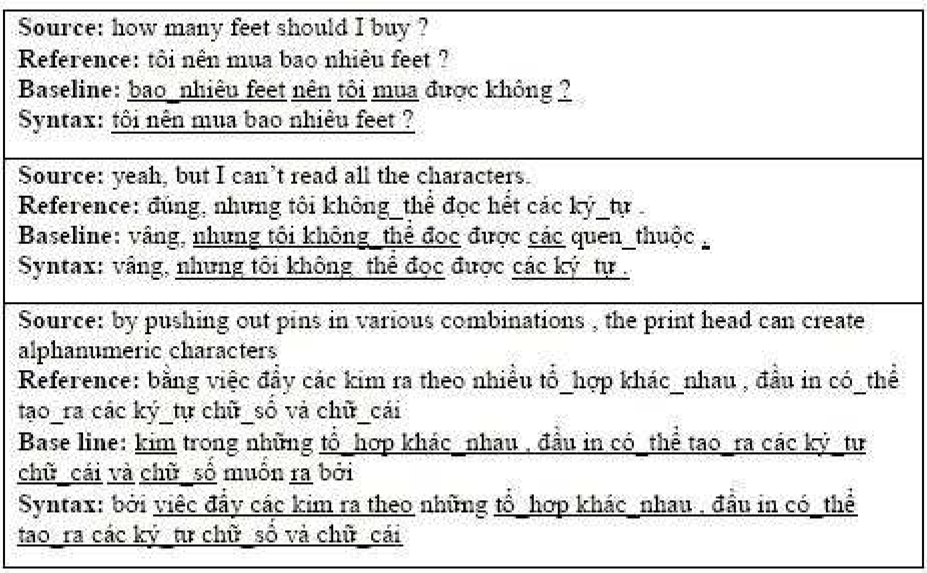


Figure 4.3: N-gram precisions



Source: how many feet should I buy ?
Reference: tôi nên mua bao nhiêu feet ?
Baseline: bao_nhiêu feet nên tôi mua được không ?
Syntax: tôi nên mua bao nhiêu feet ?

Source: yeah, but I can't read all the characters.
Reference: đúng, nhưng tôi không_thể đọc hết các ký_tự .
Baseline: vâng, nhưng tôi không_thể đọc được các quen_thuộc .
Syntax: vâng, nhưng tôi không_thể đọc được các ký_tự .

Source: by pushing out pins in various combinations , the print head can create alphanumeric characters
Reference: bằng việc đẩy các kim ra theo nhiều tổ_hợp khác_nhau , đầu in có_thể tạo_ra các ký_tự chữ_số và chữ_cái
Base line: kim trong những tổ_hợp khác_nhau , đầu in có_thể tao_ra các ký_tư chữ_cái và chữ_số muốn ra bởi
Syntax: bởi viêc đẩy các kim ra theo những tổ_hợp khác_nhau , đầu in có_thể tao_ra các ký_tư chữ_số và chữ_cái

Figure 4.4: Some examples of better translations

transformed by removing the wh-constituent from its SBARQ father to replace its co-indexed null element. For example:

(SBARQ *(WHNP-1 (WP what))* (SQ (AUX 's) (NP (DT the) (NN baby))
(VP (VBG doing) *(NP (-NONE- *-1))*)) (. ?))
→ (SBARQ (SQ (AUX 's) (NP (DT the) (NN baby)) (VP (VBG doing)
(WHNP-1 (WP what)))) (. ?))

We used this "rule-based" technique in combination with normal syntactic transformation on the Conversation corpus. The BLEU score improved from 38.12% to 38.98% (the baseline score was 35.47%). This example suggests that there is room for improving translation quality by dealing with non-local transformation.

Table 4.9: Effect of maximum phrase length on translation quality (BLEU score).

Maximum phrase size	2	3	4	5	6
Pharaoh	21.71	24.84	25.74	26.19	26.41
Syntactic transformation	24.1	27.01	27.74	27.88	28.02

Table 4.10: Effect of training-set size on translation quality (BLEU score).

Training-set size	10K	20K	40K	80K	94K
Pharaoh	21.84	23.35	24.43	25.43	25.74
Syntactic transformation	23.65	25.67	26.86	27.52	27.74

4.4.6 Maximum Phrase Length

Table 4.9 displays the performances of the baseline SMT system and the syntactic-transformation SMT system, with various maximum phrase lengths.[9] Obviously, the translation quality of both systems improves when the maximum phrase length increases. The second system can achieve high performance with a short maximum phrase length, while the first system requires a longer maximum phrase length to achieve a similar performance. The improvement of the SMT system with syntactic transformation over the baseline SMT system decreases slightly when the maximum phrase length increases. This experiment leads to two suggestions. First, a maximum phrase length of three or four is enough for the SMT system with syntactic transformation. Second, the baseline SMT system relies on long phrases to solve the word order problem, while the other SMT system uses syntactic transformation to do that.

4.4.7 Training-Set Size

In this section, we report BLEU scores and decoding times corresponding to various sizes of training sets (in terms of sentence pairs). In this experiment, we used Europarl data sets, and we chose a maximum phrase length of four. Table 4.10 shows an improvement in BLEU score of about 2% for each training set. It means the improvement over Pharaoh does not decrease significantly as the training set scales up. Note that studies which use morphological analysis for SMT have a property of vanishing improvement [Goldwater & McClosky (2005)]. Table 4.11 shows that, for all training sets, the decoding time of the SMT system with syntactic transformation is about 5-6% that of the Pharaoh system. This is an advantage of monotone decoding. Therefore we save time for syntactic analysis and transformation.

4.5 Conclusion

We have presented an approach to incorporate linguistic analysis including syntactic parsing and morphological analysis into SMT. By experiments, we have demonstrated that

[9] We used Europarl data sets.

Table 4.11: Effect of training-set size on decoding time (seconds/sent).

Training-set size	10K	20K	40K	80K	94K
Pharaoh	1.98	2.52	2.93	3.45	3.67
Syntactic transformation	0.1	0.13	0.16	0.19	0.22

this approach can improve phrase-based SMT significantly. For syntactic transformation, we employed the transformational model proposed in the previous chapter. Our method can be applied to language pairs in which the target language is poor in resources.

We have carried out various experiments with English-Vietnamese and English-French language pairs. For English-Vietnamese translation, we have shown that the combination of morpho-syntactic transformation can achieve a better result than can be obtained with either individually. On the Europarl corpus, within the range of data size with which we have experimented, we have found out that improvements made by syntactic transformation over Pharaoh do not change significantly as the corpus scales up. Moreover, a phrase-based SMT system with syntactic transformation needs shorter maximum phrase length to achieve the same translation quality as the conventional phrase-based system.

In the future, we would like to apply this approach to other language pairs, in which the differences in word order are greater than those of English-Vietnamese and English-French.

Chapter 5

Chunking-Based Reordering

According to our observation, most of studies on SMT reordering problem focus on the decoding phase, which is obviously a more natural approach than preprocessing. In this section we present a chunking-based reordering method for phrase-based SMT [Nguyen et al. (2007)]. We employ the syntactic transformation model for phrase reordering within chunks. Besides, transformation probability is also used as distortion score of translation hypotheses. There are several additional steps in the decoding phase in comparison with Pharaoh decoder [Koehn (2004)]. First, an input sentence is split into syntactic chunks using a CRFs-based chunking tool [1]. Second, a phrase graph encoding phrase-order information is built. The transformational model is used in this step. Third, the best translation sentence is generated using a beam search decoding algorithm. This algorithm, which is a version of Koehn's algorithm, employs chunking-based reordering (or specifically employs the phrase graph). This section is organized as follows: Section 4.1 introduces phrase graph structure and construction. Section 4.2 describes our phrase-based decoder. Section 4.3 reports our experimental results.

5.1 Creating a Phrase Graph

A phrase graph encodes phrase order information in its paths. Vertices represent source phrases and contain syntactic transformation probabilities. Arcs represent order relation between phrases. When a phrase graph being created, for each chunk, every possible segmentation (into phrases) will be considered. A chunk segmentation is reordered using the syntactic transformation model. Transformation probabilities are stored in vertices of that segmentation. A special treatment is required for phrases which overlap chunks. In that case, sub-chunks are generated. They are handled similarly to normal chunks.

Table 5.1 shows analyses of an example sentence. The sentence is analyzed using POS tagging, chunking, and phrase-table looking up. Then its phrase graph is generated (Figure 5.1). This graph has two overlap phrases in dotted ovals. There is a sub-chunk which is a noun phrase. Then each path, which corresponds to a chunk segmentation,

[1] http://crfpp.sourceforge.net/

Source sentence	I hear Fred is a very good student in your class
Words and phrases	I, hear, is, a, very, good, student, your, class I hear, is a, very good, good student, your class
Chunks	I | hear | Fred | is | a very good student | in | your class
Chunks (full tag)	(NP (PRP I)) (VP (VBP hear)) (NP (Fred)) (VP (VBZ is)) (NP (DT a) (RB very) (JJ good) (NN student)) (PP (IN in)) (NP (PRP$ your) (NN class))

Table 5.1: Analyses of an example sentence

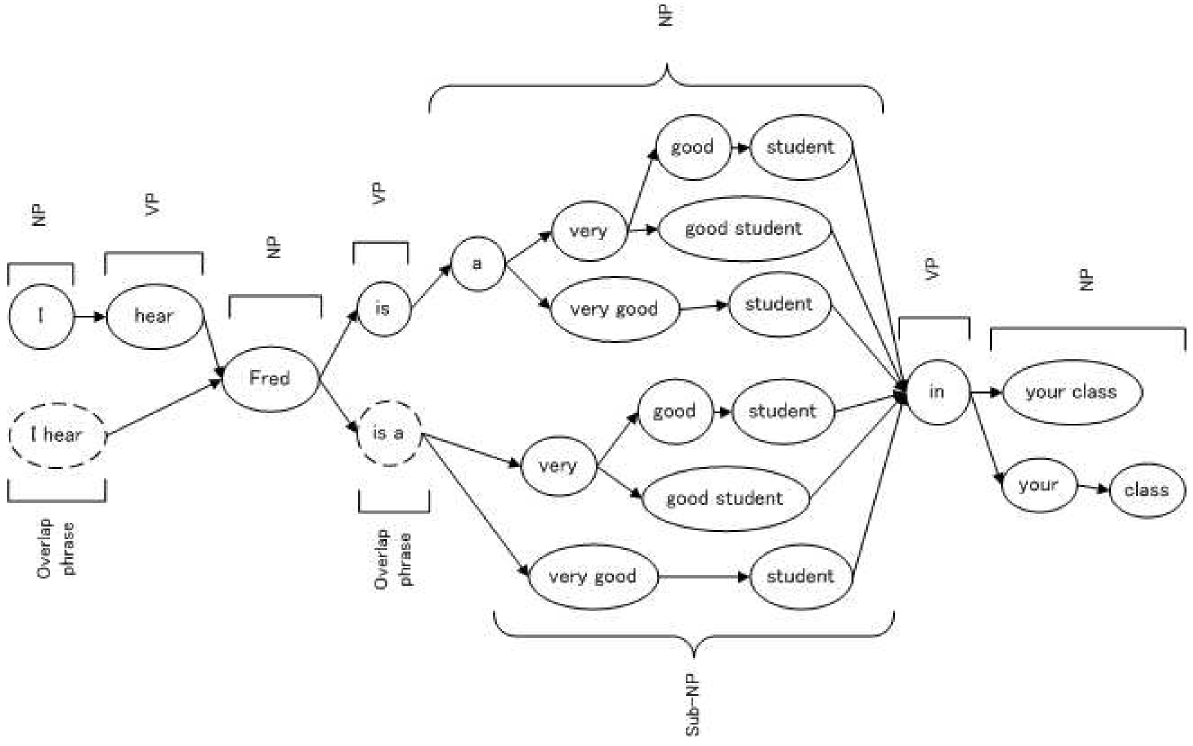


Figure 5.1: A phrase graph before reordered

will be reordered. Figure 5.2 displays reordered segmentations of the chunk ”a very good student”.

Now we describe reordering a chunk segmentation in details. First, a syntactic tree is generated and then used as the input of the transformational model. The output is a pair of the best reordering sequence and a normalized transformational probability. For example, if the input [2] is (NP (DT a) (RB very) (JJ good) (NN student)), a possible output is (0 3 1 2, -0.15). Then the reordering sequence is used to reorder the corresponding chunk segmentation (for example, ”a | very | good | student” becomes ”a | student | very | good”). The transformational probability is assigned to the last vertex of the chunk segmentation (see Figure 5.2) as its distortion score. The other vertices are assigned a zero distortion score. For segmentations which contain only one vertex, its score will be set to zero.

[2] For segmentations which contains multi-word phrases, we used the first word of a phrase for building the input syntactic tree. For example, ”very” is used instead of ”very good”.

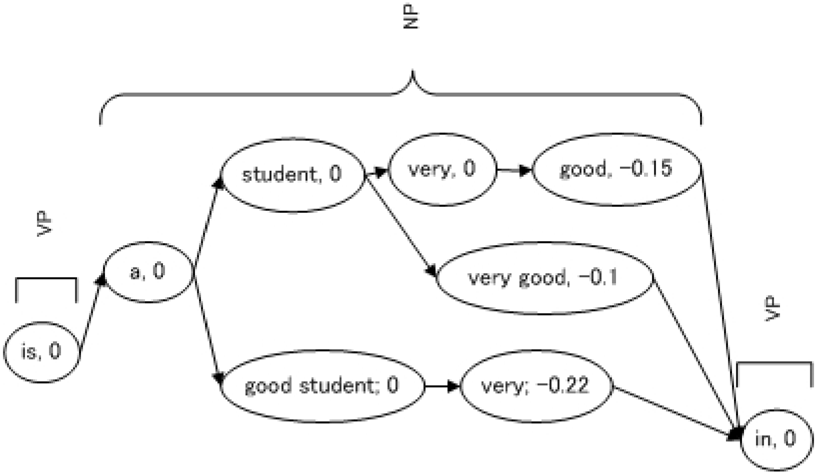


Figure 5.2: A reordered subgraph

5.2 Decoder

We implemented a beam search decoder for phrase-based log-linear translation models which uses eight feature functions:

- phrase translation score (2)
- lexical translation score (2)
- trigram language model
- target word penalty
- phrase penalty
- distortion score (movement distance or chunking based)

We adapted the beam search decoding algorithm [Koehn (2004)] for chunking-based reordering. Each translation hypothesis is associated with a vertex of the phrase graph. In state expansion, only translation options which correspond to the adjacent vertices of the current state are considered.

5.3 Experiments

We used a chunking program which is based on CRFs [Sha and Pereira (2003)]. We used entire Conversation corpus. On the Europarl corpus, we used the same test set as the previous chapter (1122 sentences) and a training set containing 20K sentences. The other experimental settings are the same as those of the previous chapter. Our decoder's baseline scores are slightly higher than Pharaoh's. The fourth column shows that chunking improves translation quality. The speed is also an advantage of chunking-based reordering. The translation time (including POS tagging, chunking, syntactic transformation, and

Table 5.2: BLEU score comparisons.

Corpus	Pharaoh	Our decoder			
		Baseline	CB reordering	Phrase restriction	Preprocessing
Conversation (En-Vn)	35.47	35.66	36.85	34.79	38.36
Europarl (En-Fr)	23.35	23.87	25.23	24.44	25.88

Table 5.3: Phrase table size comparison.

Corpus	PT size	PT size (preprocessed)	Increasing
Conversation (En-Vn)	162,289	191,995	18%
Europarl (En-Fr)	327,893	406,197	24%

decoding) reduced several times on Conversation test set and more than ten times on Europarl test set. If we only used phrases which either fully cover chunks or belong to a chunk (in Figure 5.1, the phrase "is a" partially covers the chunk "a very good student"), the translation quality downs (see the fifth column)[3]. The final column shows that preprocessing achieves the best performance. However note that in preprocessing we used deep parsing (the same as the previous chapter).

In graph construction, n-best reordering for chunks can be generated instead of one-best. The best order will be chosen by exploiting various kinds of information (in decoding). This technique can help in cases the one-best order returned by the transformational model is incorrect. We found improved examples when testing on the development set.

Now we would like to discuss about differences between using reordering in preprocessing and in decoding. There are several possible reasons for the better performance of preprocessing technique. First, since the number of cross word alignment is reduced, the phrase table size increases (Table 5.3). This property is expected to improve word choice performance. Second, by using preprocessing, a SMT system meets (or more close to) the phrase segmentation assumption (which supposes that phrase segmentations are generated under a uniform distribution [Koehn et al. (2003)]). Third, we used full-tree syntactic transformation for preprocessing which covers more syntactic structures than shallow reordering based on chunks.

5.4 Two-phase Decoding

5.4.1 Limitation of the Proposed Technique

Chunks are shallow syntactic structures, so reordering within chunks can cover only a number of phenomena. For example, in English to Vietnamese translation, base noun phrase transformation can be handled well at chunk level, but not what-question. Another reason is that, since POS tagging and chunking are not perfect, reordering can be affected at a certain degree. The definition of chunks is also a problem. For example, an

[3] This effect is similar to that of Koehn's experiments [Koehn et al. (2003)] when he restricted phrases to syntactic ones

English prepositional chunk does not contain its corresponding noun phrase. So if we are considering English-Japanese language pair, an English preposition will not be moved to the end of a phrase to become a postposition. For these reasons, at the conceptual level, reordering over-chunk is necessary.

5.4.2 Two-phase Decoding

We propose a simple procedure to carry out over-chunk reordering. This procedure has two phases. In the first phase, chunks are translated. In the second phase, the whole sentence is translated based on chunk translations. Chunk translation is based on phrase graph. Both chunks and sub chunks are translated. Reordering is controlled by syntax and within chunk. Distortion score is based on transformational probability. Decoding algorithm is mentioned in Section 2. Resulting chunk translations will be considered as translation options of the next phase. Decoding in the second phase can simply be a normal decoding procedure. We use the baseline distortion model which is based on movement distance[4]. Our experiments on the two data sets did not show a significant change in BLEU score: 36.7 (Conversation) and 25.33 (Europarl).

5.5 Conclusion

Chunk-based reordering can improve both translation quality and translation speed. Currently, our method only works with shallow syntactic structure (only one level). We would like to extend this approach for deeper syntactic structure (multi level).

Recently, [Zhang et al. (2007)] proposed a method in which chunk reordering rules were used in preprocessing phase to create a reordered word lattice of a source sentence. Then in the next phase, there was no reordering, only monotonic decoding was enough. [Zhang et al. (2007)]'s method can be considered as a kind of preprocessing. A number of important differences from our work presented in the previous chapter include: shallow syntactic structure vs. deep syntactic structure, n-best vs. 1-best, and language model scoring of word lattice vs. syntactic transformation model scoring of phrase lattice. [Costa-jussia et al. (2007)] proposed a similar method with [Zhang et al. (2007)]'s for n-gram based SMT.

[4] Of course, other models can be used such as lexicalized models [Koehn & Hoang (2007)] or POS tag based models [Quirk et al. (2005)]

Chapter 6

Syntax-Directed Phrase-Based SMT

In the previous chapter, we presented a chunking-based reordering method that works on shallow syntactic structures. This chapter describes a general framework for tree-to-string phrase-based SMT. This frame work is based on a form of stochastic syntax-directed translation schemata [Aho & Ullman (1972)]. Our experimental results of English-Japanese and English-Vietnamese translation showed a significant improvement over a baseline phrase-based SMT system.

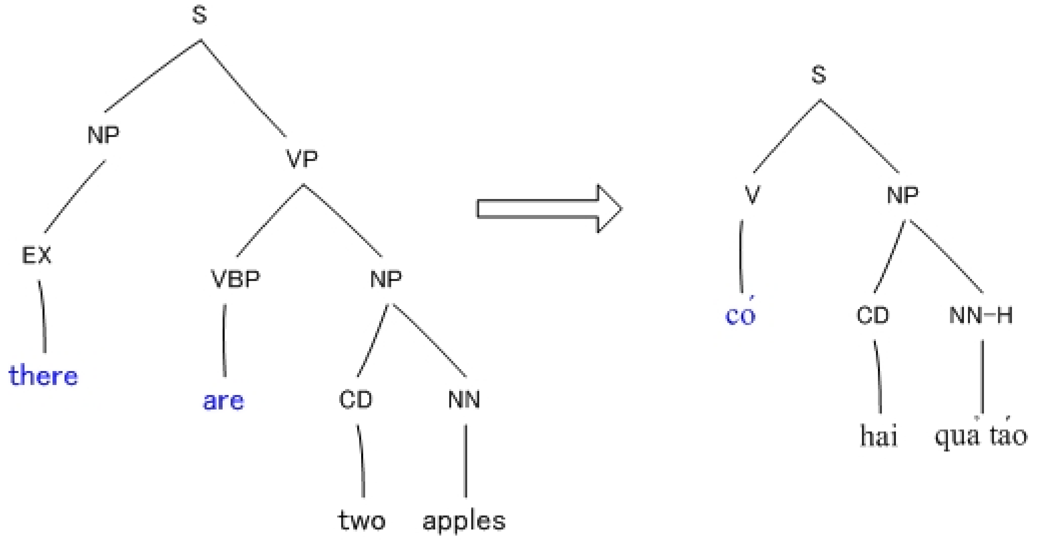


Figure 6.1: Non-constituent phrasal translation (English-Vietnamese).

There are a number of motivations behind this work. First, from our point of view, it is important to make available to syntax-based models all the bilingual phrases that are typically available to phrase-based models. Figure 6.1 shows an example of non-constituent phrasal translation. Second, we would like to use the syntactic transformation model mainly as a reordering model in decoding phase without the limitation of shallow syntactic structures. And third, we would like use a PCFG as a distribution over phrase segmentations instead of the uniform distribution [Koehn et al. (2003)].

6.1 A Stochastic Syntax-Directed Translation Schema for Phrase-Based SMT

6.1.1 Stochastic Syntax-Directed Translation Schema

A stochastic syntax-directed translation schema (SSDTS) is a 5-tuple $T = (N, \Sigma, \Delta, R, S)$ where

- N = a finite set of nonterminals;
- Σ = a finite input alphabet;
- Δ = a finite output alphabet;
- R = a finite set of rules of the form $p : A \rightarrow \alpha, \beta$ for A in N, α in $(N \cup \Sigma)^*$, β in $(N \cup \Delta)^*$, $0 < p \leq 1$, with the nonterminals in α being a permutation of those in β;
- S in N = the starting symbol.

Since we want to apply this schema to phrase-based SMT, we consider a kind of SSDTS whose rules have two following forms:

- $p : A \rightarrow \alpha, \beta$ for A in N, α in N^*, β in N^*, $0 < p \leq 1$, with the nonterminals in α being a permutation of those in β. Note that this kind of rule contains only nonterminal symbols on the right hand side.
- $p : B \rightarrow \gamma, \delta$ for B in N, γ in Σ^*, δ in Δ^*, $0 < p \leq 1$, γ is a phrase in source language and δ is its translation in target language.

A SSDTS has two associated context-free grammars: a source grammar $G_s = (N, \Sigma, R_s, S)$ and a target grammar $G_t = (N, \Delta, R_t, S)$. These grammars have the same set of nonterminals and the same start symbol. Their rules are paired together by rules of SSDTS. There are two kind of rules: one generating a sequence of nonterminals and the other generating a phrase.

6.1.2 A Tree-to-String SMT Model

Now we describe a tree-to-string SMT model based on SSDTS. The translation process is shown in 6.1:

$$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow S \tag{6.1}$$

where T_1 is a source tree, T_2 is a source phrase tree, T_3 is a target phrase tree, and S is a target string.

Using the first order chain rule, the probability of a target string is calculated as in 6.2:

$$P(S|T_1) = P(T_1) \times P(T_2|T_1) \times P(T_3|T_2) \times P(S|T_3) \tag{6.2}$$

$P(T_1)$ can be omitted since only one syntactic tree is used. $P(T_2|T_1)$ is a word-to-phrase tree transformation model we describe later. $P(S|T_3)$ can be calculated using a language model $P_{lm}(S)$. $P(T_3|T_2)$ is computed using SSDTS:

$$P(T_3|T_2) = \prod P(A \rightarrow \alpha, \beta) \times \prod P(B \rightarrow \gamma, \delta) \tag{6.3}$$

The first term is a reordering model and the second is a phrase translation model. So we can write:

$$S^* = \arg\max_S [P(T_2|T_1) \times \prod P(A \rightarrow \alpha, \beta) \times \prod P(B \rightarrow \gamma, \delta) \times P_{lm}(S)] \tag{6.4}$$

The translation equation 6.4 contains familiar components including a syntax-based reordering model, a phrase translation model, and a language model. A new component is the word-to-phrase tree transformation model. This is the fundamental equation of our study represented in this chapter. In the next section, we will describe how to transform a word-based CFG tree into a phrase-based CFG tree.

6.2 Transformation of a CFG Tree into a Phrase CFG Tree

6.2.1 Penn Treebank's Tree Structure

According to this formalism, a tree is represented by phrase structure. If we extract a CFG from a tree or set of trees, there will be two possible rule forms:

- $A \rightarrow \alpha$ where α is a sequence of nonterminals (syntactic categories).
- $B \rightarrow \gamma$ where γ is a terminal symbol (or a word in this case).

We consider an example of a syntactic tree and a simple CFG extracted from that tree.

Sentence: *"I am a student"*

Syntactic tree: *(S (NP (NN I)) (VP (VBP am) (NP (DT a) (NN student))))*

Rule set: *S → NP VP; VP → VBP NP; NP → NN | DT NN; NN → I | student;*

VBP → am; DT → a

However, we are considering phrase-based translation. Therefore the right hand side of the second rule form must be a sequence of terminal symbols (or a phrase) but not a single symbol (a word). We again consider the previous example. Suppose that the

+ Input:	A CFG tree, a phrase segmentation
+ Output:	A phrase CFG tree
+ Step 1:	Allocate phrases to leaf nodes. A phrase is allocated to head word of a node if the phrase contains the head word. This head word is then considered as the phrase head. This is a top-down procedure. It is applied to all phrases.
+ Step 2:	Transform the syntactic tree by replacing leaf nodes by their allocated phrases and removing all fully-covered nodes.

Table 6.1: An algorithm to transform a CFG tree to a phrase CFG tree.

phrase table contains a phrase *”am a student”* which leads to the following possible tree structure:

Phrase segmentation: *”I | am a student”*

Syntactic tree: *(S (NP (NN I)) (VP (VBP am a student)))*

Rule set: *S → NP VP; VP → VBP; NP → NN; NN → I; VBP → am a student*

We have to find out some way to transform a CFG tree into a tree with phrases at leaves. In the next subsection we propose such an algorithm.

6.2.2 An Algorithm for Word-to-Phrase Tree Transformation

Table 6.1 represents our algorithm to transform a CFG tree to a phrase CFG tree. When designing this algorithm, our criterion is to preserve the original structure as much as possible. This algorithm includes two steps. There are a number of notions concerning this algorithm:

- A CFG rule has a head symbol on the right hand side. Using this information, head child of a node on a syntactic tree can be determined.
- If a node is a pre-terminal node (containing POS tag), its head word is itself. If a node is an inner node (containing syntactic constituent tag), its head word is retrieved through the head child.
- Word span of a node is a string of its leaves. For instance, word span of subtree (NP (PRP$ your) (NN class)) is ”your class”.

Now we consider an example depicted in Figure 6.2 and 6.3. Head children are tagged with functional label H. There are two phrases: ”is a” and ”in your class”. After the Step 1, the phrase ”is a” is attached to (VBZ is). The phrase ”in your class” is attached to (IN in). In Step 2, the node (V is) is replaced by (V ”is a”) and (DT a) is removed from its father NP. Similarly, (IN in) is replaced by (IN ”in your class”) and the subtree NP on the right is removed.

The proposed algorithm has some properties. We state these properties without presenting proof[1].

[1] Proofs are simple.

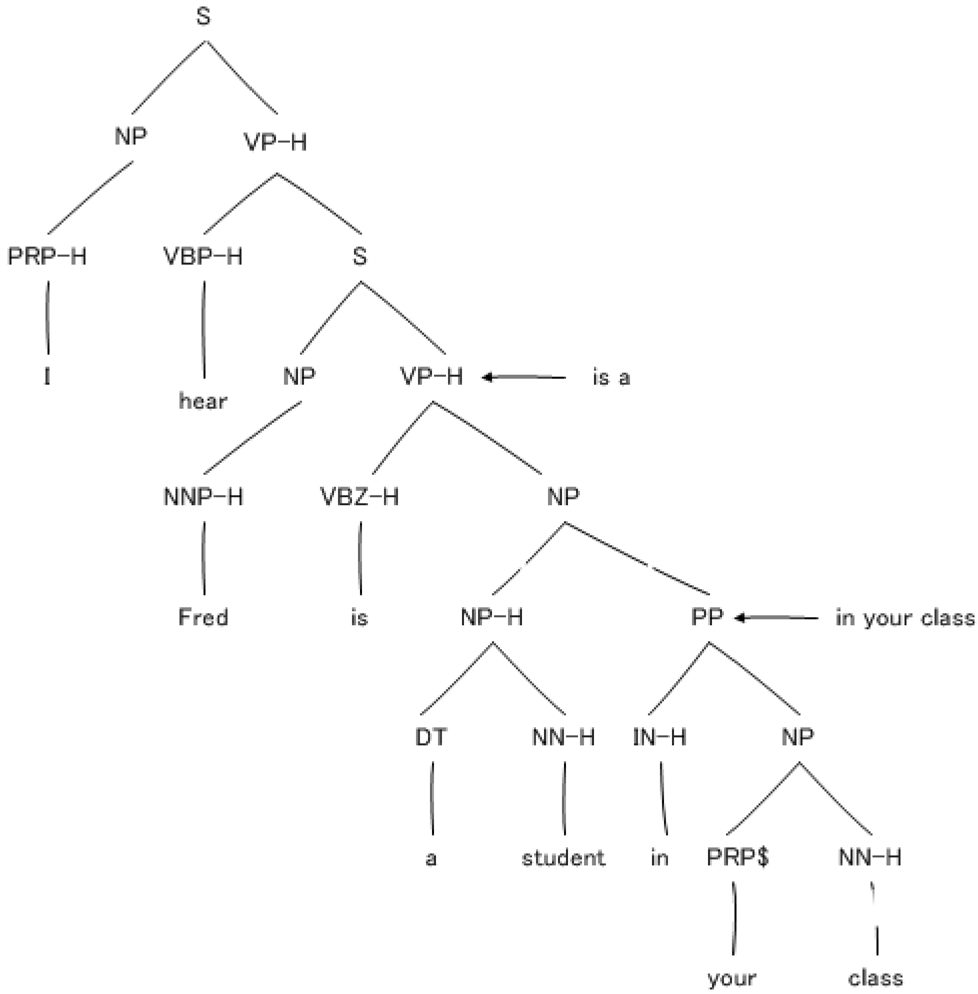


Figure 6.2: Tree transformation: Step 1.

- **Uniqueness:** Given a CFG tree and a phrase segmentation, by applying Algorithm 6.1, one and only one phrase tree is generated.
- **Constituent subgraph:** A phrase CFG tree is a connected subgraph of input tree if leaves are ignored.
- **Flatterness:** A phrase CFG tree is flatter than input tree.
- **Outside head:** The head of a phrase is always a word whose head outside the phrase. If there is more than one word satisfying this condition, the word at the highest level is chosen.
- **Dependency subgraph:** Dependency graph of a phrase CFG tree is a connected subgraph of input tree's dependency graph if there exist no detached nodes.

The meaning of Property 1 is that our algorithm is a deterministic procedure. Property 2 will be employed in the next section for an efficient decoding algorithm. When a syntactic tree is transformed, a number of subtrees are replaced by phrases. The head word of a phrase is the contact point of that phrase with the remaining part of a sentence.

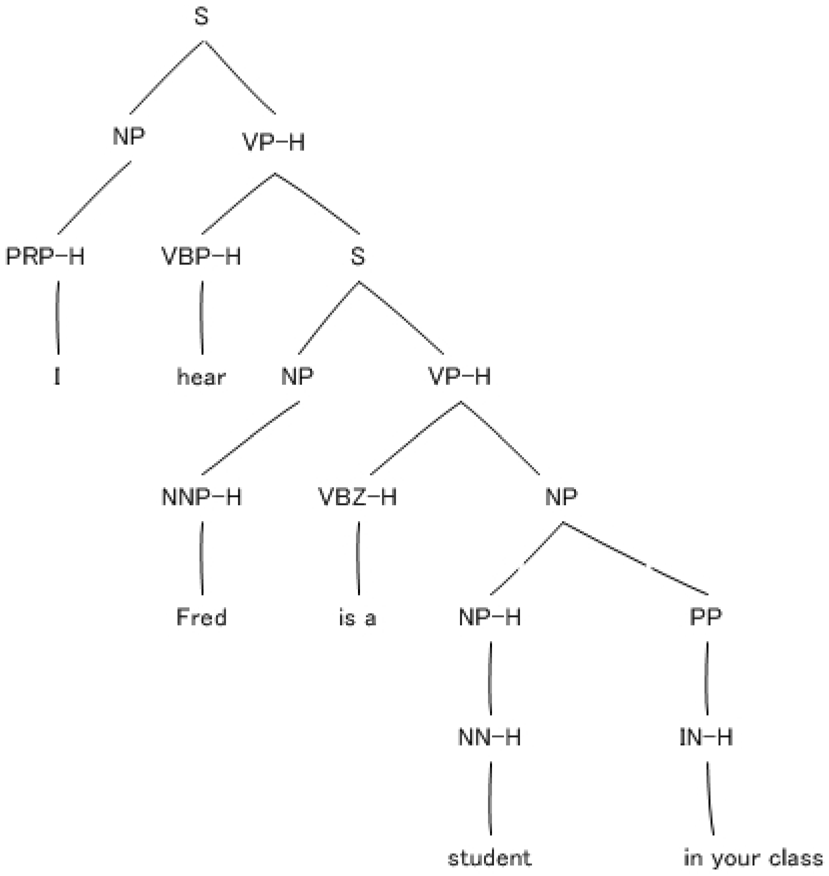


Figure 6.3: Tree transformation: Step 2.

From the dependency point of view, head word depends on an outer word is better than an inner word. About Property 5, when there is a detached node, an indirect dependency will become a direct one. In any cases, there is no change in dependency direction. We can observe dependency trees in Figure 6.4. The first two trees are source dependency tree and phrase dependency tree of the previous example. The last one corresponds to the case in which a detached node exists.

6.2.3 Probabilistic Word-to-Phrase Tree Transformation

We have proposed an algorithm to create a phrase CFG tree from a pair of CFG tree and phrase segmentation. Two questions naturally arise: ”is there a way to evaluate how good a phrase tree is?” and ”is such an evaluation valuable?” Note that a phrase tree is the means to reorder the source sentence represented as a phrase segmentation. Therefore a phrase tree is surely not good if there is no right order can be generated. Now the answer to the second question is clear. We need an evaluation method to prevent our program from generating bad phrase trees. In other words, good phrase trees should be given a higher priority.

We consider a linguistically motivated example of English-Japanese translation. This example shows the main problem of word-to-phrase tree transformation. We use partial

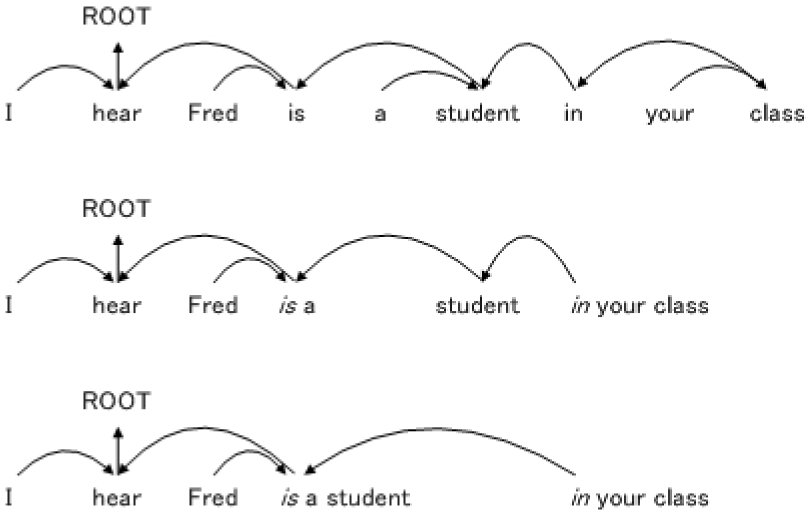


Figure 6.4: Dependency trees.

tree representation. Note that in phrase-based SMT, a translation option is a phrase pair. When the translation option whose syntactic patterns are NN-H PP and PP NN-H is chosen, there is no way to generate the expected target translation.

Test sentence: (S NP (VP-H VB-H (NP DT NN NN-H) PP))

Translation option: NN-H PP, PP NN-H

Expected target: (S NP (VP-H PP (NP DT NN NN-H) VB-H))

Incorrect-word-order output: (S NP (VP-H (NP DT NN *PP NN-H*) VB-H))

Tree transform: (S NP (VP-H VB-H (NP DT NN NN-H) PP))

$\Longrightarrow$ (S NP (VP-H VB-H (NP DT NN NN-H+)))

where NN-H+ is NN-H PP

Problematic transformation: NP $\rightarrow$ DT NN NN-H $\Longrightarrow$ NP $\rightarrow$ DT NN NN-H+

Now we consider phrase trees in the context of training phase of phrase-based SMT. In this phase, all phrase pairs that are consistent with the word alignment are collected. Consistency with word alignment is dependent on context. Figure 6.5 shows an example. A phrase tree is acceptable if its phrases are consistent with word alignment. Therefore given a sentence pair, a word alignment, and a syntactic tree, all possible phrase trees are acceptable. This observation suggest us a way to compute phrase tree probability. In 6.5, we define the phrase tree probability as the product of its rule probability given the original CFG rules. Conditional probabilities are computed in a separate training phase using a source-parsed and word-aligned bitext.

$$P(T') = \prod_i P(LHS_i \rightarrow RHS'_i | LHS_i \rightarrow RHS_i) \tag{6.5}$$

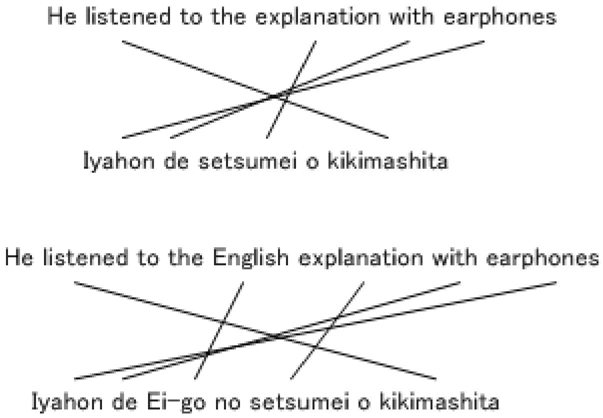

He listened to the explanation with earphones
Iyahon de setsumei o kikimashita
He listened to the English explanation with earphones
Iyahon de Ei-go no setsumei o kikimashita

Figure 6.5: The phrase pair (”explanation with earphones”, ”iyahon de setsumei”) is consistent with the word alignment in the first sentence pair but it can not be applied to translate the second source sentence.

where T' is a phrase tree whose CFG rules are $LHS_i \rightarrow RHS'_i$. $LHS_i \rightarrow RHS'_i$ are original CFG rules. RHS'_i are subsequences of RHS_i. Since phrase tree rules should capture changes made by the transformation from word to phrase, we use ’+’ to represent an expansion and ’-’ to show an overlap. These symbol will be added to a nonterminal on the side having a change. In the previous example, since a head noun in the word tree has been expanded on the right, the corresponding symbol in phrase tree is NN-H+. A nonterminal X can become one of the following symbols $X, -X, +X, X-, X+, -X-, -X+, +X-, +X+$.

6.3 Decoding

A source sentence can have many possible phrase segmentations. Each segmentation in combination with a source tree corresponds to a phrase tree. A phrase-tree forest is a set of those trees. A naive decoding approach is that for each segmentation, a phrase tree is generated and then the sentence is translated. This approach is very slow or even intractable. Based on the Property 2 of the tree transformation algorithm, the forest of phrase trees will be packed into a tree-structure container whose backbone is the original CFG tree.

6.3.1 Translation Options

A translation option encodes a possibility to translate a source phrase (at a leaf node of a phrase tree) to another phrase in target language. Since our decoder uses a log-linear translation model, the decoder can exploit various features of a translation option. We use the same features as [Koehn et al. (2003)]. Basic information of a translation option includes:

- source phrase

- target phrase
- phrase translation score (2)
- lexical translation score (2)
- word penalty

Translation options of an input sentence are collected before any decoding takes place. This allows a faster lookup than consulting the whole phrase translation table during decoding. Note that the entire phrase translation table may be to big to fit into memory.

6.3.2 Translation Hypotheses

A translation hypothesis represents a partial or a full translation of an input sentence. Initial hypotheses correspond to translation options. Each translation hypothesis is associated with a phrase-tree node. In other words, a phrase-tree node has a collection of translation hypotheses. Now we consider information contained in a translation hypothesis:

- the cost so far
- list of child hypotheses
- left language model state and right language model state

6.3.3 Decoding Algorithm

First we consider structure of a syntactic tree. A tree node contains fields such as syntactic category, child list, and head child index. A leaf node has an additional field of word string. In order to extend this structure to store translation hypotheses, a new field of hypothesis collection is appended. A hypothesis collection contains translation hypotheses whose word spans are the same. Actually, it corresponds to a phrase-tree node. A hypothesis collection whose word span is $[i_1, i_2]$ at a node whose tag is X expresses that:

- There is a phrase-tree node (X, i_1, i_2).
- There exist a phrase $[i_1, i_2]$ or
- There exist a subsequence of X's child list: (Y_1, j_0, j_1), $(Y_2, j_1 + 1, j_2)$, ..., $(Y_n, j_{n-1} + 1, j_n)$ where $j_0 = i_1$ and $j_n = i_2$
- Suppose that $[i, j]$ is X's span, then $[i_1, i_2]$ is a valid phrase node's span if and only if: $i_1 <= i$ or $i < i_1 <= j$ and there exist a phrase $[i_0, i_1 - 1]$ overlapping X's span at $[i, i_1 - 1]$. A similar condition is required of j.

+ Input:	A source CFG tree, a translation-option collection
+ Output:	The best target sentence
+ Step 1:	Allocate translation options to hypothesis collections at leaf nodes.
+ Step 2:	Compute overlap vector for all nodes.
+ Step 3:	For each node, if all of its children have been translated, then for each valid sub-sequence of child list, carry out the following steps:
+ Step 3.1:	Generate a lexicalized CFG rule and find the best order in the target language
+ Step 3.2:	Reorder the sub-sequence
+ Step 3.3:	Translate the reordered sub-sequence and update corresponding hypothesis collections

Table 6.2: Decoding algorithm.

Input sentence:	he listened to the English explaination
Input tree:	Figure 6.6
Translation options:	he|””; listened to|kikimashita; the|””; English explanation|Ei-go no setsumei o; English|Ei-go; explanation|setsumei o
Possible outputs:	Ei-go no setsumei o kikimashita; Ei-go setsumei o kikimashita

Table 6.3: An example of English-Japanese translation.

Table 6.2 shows our decoding algorithm. Step 1 distributes translation options to leaf nodes using a procedure similar to Step 1 of algorithm 6.1. Step 2 helps check valid subsequences in Step 3 fast. Step 3 is a bottom-up procedure, a node is translated if all of its child nodes have been translated. Step 3.1 calls a syntactic transformation model with input parameter a lexicalized CFG rule. After reordered in Step 3.2, a subsequence will be translated in Step 3.3 using a simple monotonic decoding procedure resulting in new translation hypotheses.

We consider an example of translation from English to Japanese in Table 6.3. There are two possible phrase trees depicted in Figures 6.7 and 6.8. Each phrase tree results in a Japanese sentence. Table 6.4 shows how the decoding algorithm is applied to this example. This process generates the same two output sentences.

Node	Child sequence	LCFG rule	Order	Translation
NP	PRP[0]	(NP (PRP he))	0	””
NP	NN[3,4]	(NP (NN explanation))	0	Ei-go no setsumei o
	JJ[3] NN[4]	(NP (JJ English) (NN explanation))	0 1	Ei-go setsumei o
VP	VBD[1,2] NP[3,4]	(VP (VBD listened) (NP(NN explanation)))	1 0	Ei-go no setsumei o kikimashita ; Ei-go setsumei o kikimashita
S	NP[0] VP[1,4]	(S (NP (PRP he)) (VP (VBD listened)))	0 1	Ei-go no setsumei o kikimashita ; Ei-go setsumei o kikimashita

Table 6.4: Bottom-up translation.

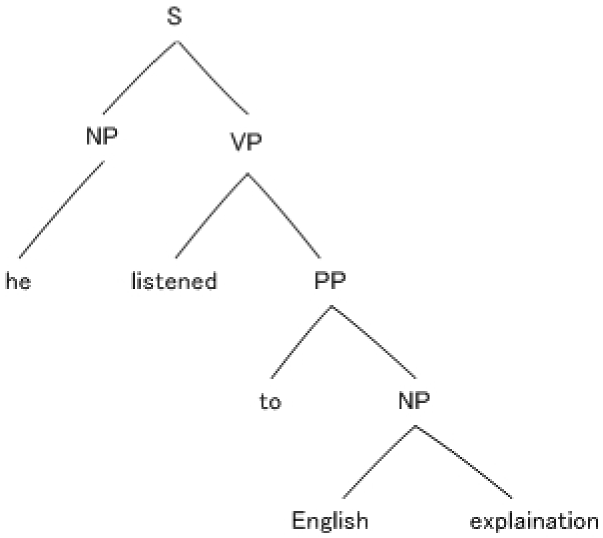


Figure 6.6: English source tree.

		English		Japanese
Training	Sentences		55,758	
	Average sentence length	26.7		33.5
	Words	1,488,572		1,867,952
	Vocabulary	31,702		29,406
Test	Sentences		1,021	
	Average sentence length	26.43		33.32
	Words	26,980		34,023
	Vocabulary	4,628		4,348

Table 6.5: Corpus statistics of English-Japanese translation task.

6.4 Experimental Results

We used Reuters[2], an English-Japanese bilingual corpus. This corpus was split into two data sets as shown in Table 6.5. Japanese sentences were analyzed by ChaSen[3], a word-segmentation tool. Another corpus is Conversation which has been described in Chapter 4.

Table 6.4 shows a comparison of BLEU scores between Pharaoh, our phrase-based

[2] http://www2.nict.go.jp/x/x161/members/mutiyama/index.html

[3] http://chasen.aist-nara.ac.jp/chasen/distribution.html.en

Corpus	Pharaoh	Our PB system	Our CBR system	Our SD system
Conversation	35.47	35.66	36.85	37.42
Reuters	24.41	24.20	20.60	25.53

Table 6.6: BLEU score comparison between phrase-based SMT and syntax-directed SMT. PB=phrase-based; CBR=chunk-based reordering; SD=syntax-directed

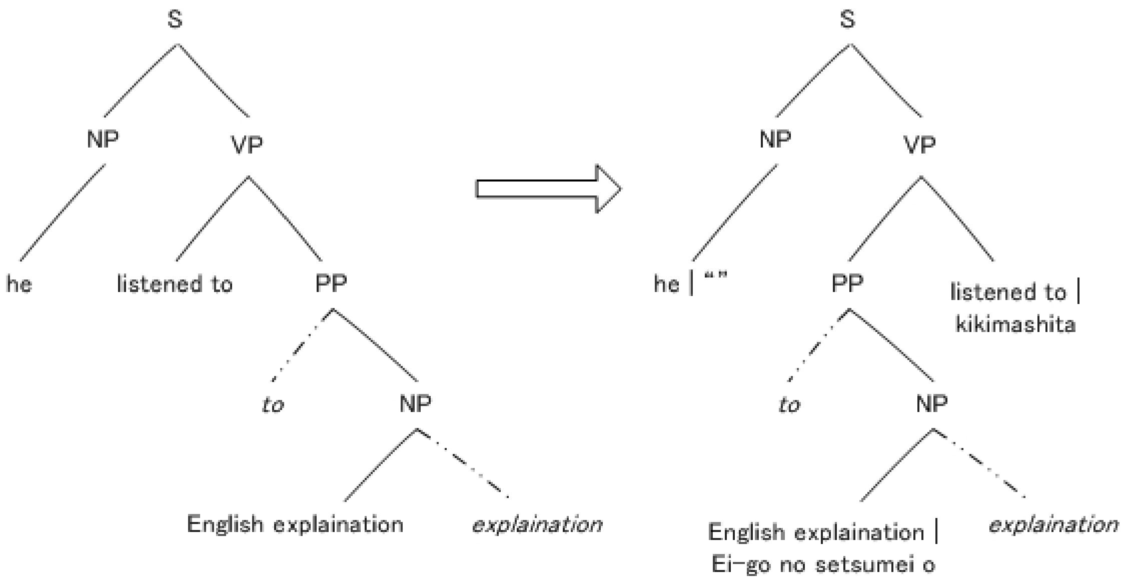


Figure 6.7: Translation according to phrase tree 1.

SMT system, our chunk-based-reordering SMT system, and our syntax-directed SMT system. On both Conversation corpus and Reuters corpus: The BLEU score of our phrase-based SMT system is comparable to that of Pharaoh; The BLEU score of our syntax-directed system is higher than that of our phrase-based system. On Conversation corpus, our chunk-based reordering system has a higher performance in terms of BLEU score than our phrase-based system. Using sign test [Lehmann (1986)], we verified the improvements are statistically significant. However, on Reuters corpus, performance of the chunk-based reordering system is much lower than the phrase-based system's. The reason is that in English-Japanese translation, chunk is a too shallow syntactic structure to capture word order information. For example, a prepositional chunk often includes only preposition and adverb, therefore such information does not help reordering prepositional phrases.

6.5 Conclusions

We have presented a general tree-to-string phrase-based approach. This approach employs a syntax-based reordering model in the decoding phase. By word-to-phrase tree transformation, all possible phrases are considered in translation. Our approach does not suppose a uniform distribution over all possible phrase segmentations as [Koehn et al. (2003)] since each phrase tree has a probability. We believe that by using n-best trees, translation quality can be improved further. A number of non-local reordering phenomena such as adjunct attachment should be handled in the future.

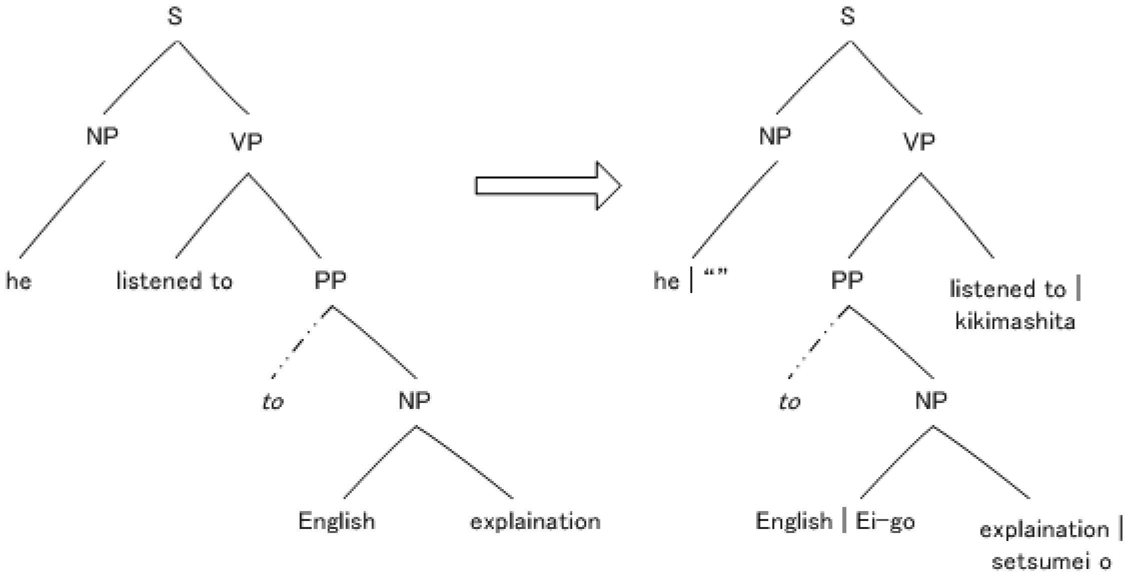


Figure 6.8: Translation according to phrase tree 2.

Chapter 7

Integration of Word Sense Disambiguation into Phrase-Based Statistical Machine Translation

Beside the word order problem, word choice is another obstacle for MT. Though phrase-based SMT has an advantage of word choice based on local context, exploiting larger context is an interesting research topic. We apply principles and techniques from word sense disambiguation (WSD) studies to phrase-based SMT. WSD score is used as a feature of translation. Besides we investigate how to train WSD models for this application. We also analyze properties of this kind of WSD in comparison with standard WSD tasks. Recently, [Chan et al. (2007)] and [Carpuat and Wu (2007)] have showed that WSD can improve SMT significantly. Basically, our method is similar to that of [Carpuat and Wu (2007)]. The differences include: we use MEM and NB classifiers; we evaluate the WSD accuracy; and we analyze effects of phrase-length limitation and the use of syntactic relation feature.

7.1 Introduction

In this chapter, we present our study on this topic. First, we give some background knowledge about WSD and SMT. Then we describe how WSD is used for SMT. Next we show our experimental results. We analyze various settings of WSD-SMT integration. Our results give a thorough view into the problem.

7.2 WSD

There are many words which have more than one sense or meaning. The problem of word sense disambiguation is that given a word and its context, how to choose an appropriate sense. There are many approaches to solve this problem: corpus-based approaches (using supervised or unsupervised learning), dictionary-based approaches, and the combination of them. In this study, we focus on how to use WSD models, trained using supervised methods, for phrase-based SMT systems.

7.2.1 WSD Models

One of the most successful current lines of research is the corpus-based approach, in which statistical or machine learning algorithms have been applied to learn statistical models or classifiers from corpora in order to perform WSD. A supervised WSD system requires an annotated dataset which includes labeled (or tagged) examples, in which each example contains the target word **w** assigned with its right sense. This data, called labeled data or training data, is then used for a supervised learning algorithm to train a classifier for future detection of test examples.

Until now, many ML algorithms have been applied to WSD, such as: Decision Lists [Yarowsky (1994)], Neural Networks [Towell & Voorhees (1998)], Bayesian learning [Bruce & Wiebe (1994)], Exemplar-based learning [Ng (1997)], Boosting [Escudero *et al.* (2000b)], etc. Further, in [Mooney (1996)] some of the previous methods are compared jointly with Decision Trees and Rule Induction algorithms, on a very restricted domain. Recently, [Lee & Ng(2002)] evaluates some strong ML algorithms in WSD, including Support Vector Machines, AdaBoost, and Decision Tree. This work was accomplished for the recently competition data including Senseval-1 and Senseval-2. In the contest of Senseval-3, [Ngai *et al.* (2004)] also investigate the semantic role labeling with Boosting, SVMs, Maximum Entropy, SNOW, and Decision Lists.

Reviewing results from these studies, we can conclude that there is no ML algorithm which dominates others. This conclusion is based on two observations that there are no large distance between the accuracies obtained from different algorithms, and the best algorithm are changed via different corpora. In this study we would like to investigate the use of Nave Bayes and MEM because these models can deal with a large number of training examples, a common problem in machine translation. We will compare WSD accuracy and translation improvement made by these methods.

7.2.2 WSD Features

Context is the only means to identify the meaning of a polysemous word. Therefore, all work on sense disambiguation relies on the context of the target word to provide information to be used for its disambiguation. For corpus-based methods, context also provides the prior knowledge with which current context is compared to achieve disambiguation.

According to [Ide *et al.* (1998)], context is used in two ways:

- *The bag of words*: here, context is considered as words in some window surrounding the target word, regarded as a group without consideration for their relationship to the target in terms of distance, grammatical relations, ect.

- *Relational information*: context is considered in terms of some relations, selectional preferences, orthographic properties, phrasal collocation, semantic categories, ect.

[Ng & Lee (1996)] is considered as the first study in which various linguistic knowledge sources are used, including topical context, collocation of words, and the verb-object syntactic relationship. [Leacock (1998)] used more kinds of information, that are words or part-of-speech tags assigned with their positions in relation to the target word. More currently, [Lee & Ng(2002)] used all these kinds of information, which then become popular sources of knowledge for recent studies, but with some modifications (add or remove some knowledge sources), se [Le & Shimazu (2004), Le *et al.* (2005a), Ando(2006)].

Suppose that $\mathbf{w}$ is the polysemous word to be disambiguated, and $S = \{s_1, \dots, s_m\}$ is the set of its potential senses. Given a context W of $\mathbf{w}$ represented as:

$$W = w_{-N^L}, \dots, w_{-1}, w_0, w_{+1}, \dots, w_{+N^R} \tag{7.1}$$

W is a context of $\mathbf{w}$ within a windows $(-N^L, +N^R)$ in which $w_0 = \mathbf{w}$ is the target word; for each $i \in \{N^L, \dots, -1, +1, \dots, +N^R\}$, w_i is a word appearing at the position i in relation with $\mathbf{w}$. If $i < 0$ then w_i stands in the left of $\mathbf{w}$, and if $i > 0$ then w_i stands in the right of $\mathbf{w}$. For simplify, we assume $N^L = N^R$ and denote this value by N.

Up to now, most studies use part-of-speech information as an important knowledge source for determining word senses. Therefore, the sentences containing $\mathbf{w}$ should be POS tagged. Denote the context in (7.1) after POS tagging as:

$$W = [w_{-N}, p_{-N}], \dots, [w_{-1}, p_{-1}], [w_0, p_0], [w_{+1}, p_{+1}], \dots, [w_{+N}, p_{+N}] \tag{7.2}$$

In the below, we divide the usually used knowledge into the four kinds.

Topical Context

Topical context includes substantive words that co-occur with a given sense of the polysemous word, usually within a window of several sentences. Unlike micro-context, which has played a role in disambiguation work since the early 1950s, topical context has been less consistently used. Methods relying on topical context exploit redundancy in a text–that is, the repeated use of words that are semantically related throughout a text on a given topic. Thus, base is ambiguous, but its appearance in a document containing words such as *pitcher*, and *ball* is likely to isolate a given sense for that word (as well as the others, which are also ambiguous). Work involving topical context typically uses the bag-of-words approach, in which words in the context are regarded as an unordered set. [Yarowsky (1992)] uses a 100-word window, both to derive classes of related words and as context surrounding the polysemous target, in his experiments using Roget's Thesaurus. [Gale *et al.* (1993)], looking at a context of ±50 words, indicate that while words closest to the target contribute most to disambiguation, they improved their results from 86% to 90% by expanding context from ±6 (a typical span when only micro-context is considered) to ±50 words around the target. All studies use this kind of information as an important part of the whole knowledge used for disambiguating senses, such as [Pedersen (2000), Lee & Ng(2002)].

Topical context is represented by a set of unordered words in a certain window size. Particularly, if the context is represented as in (7.2), then a topic context in a window $(-M, +M)$ is represented by the set, denoted by TC, as follows:

$$TC = \{w_{-M}, \ldots, w_{-1}, w_{+1}, \ldots, w_{+M}\}$$

Local Words

Using "Local Words" we want to mention the information extracted from "the words in a local context". Note that a "local context" is a context containing the target word in a small size. In our opinion, *collocations* and *ordered words*, which are wildly used in WSD studies, can be grouped into this kind of information.

Collocations According to [Ide *et al.* (1998)], a significant collocation can be defined as a syntagmatic association among lexical items. With the context W as represented in (7.2), a collocation is defined as a sequence of words from the position $-l$ to the position $+r$: $w_{-l} \ldots w_0 \ldots w_r$, where $l \geq 0$, $r \geq 0$, and $l + r \geq 1$. As usually used in previous works, we design a set of collocations based on the maximum length of these collocations. Denote $ColW$ be the set of collocations of maximum length Len, it is defined as:

$$ColW = \{w_{-l} \ldots w_0 \ldots w_{+r} | l \geq 0; r \geq 0; l + r \geq 1; l + r \leq Len\}$$

Ordered Words Different to unordered words in topical context, each ordered word consists of a word and its position in relationship with the target word. In our view, ordered words in a local context contain information about semantic and syntax relations between neighbor words and the target word. The set of ordered words in a local window (-l,+r), denoted by OW, consists of pairs (w_i, i), as follows.

$$OW = \{(w_i, i) | i = -l, \ldots, +r\}$$

Local Part-Of-Speeches

Using "Local Part-Of-Speeches (POSs)" we want to mention the information extracted from "the Part-Of-Speeches tags in a local context". Similar to Local Words, the kinds of information of also consists of collocations of POSs and ordered POSs. The difference here is that the Local Word involves the orthographic forms of the neighbor words while Local POS involves their part-of-speech forms. In a window $(-l, +r)$ the set of collocation of POSs with maximum length Len, denoted by $ColP$, and the set of ordered POSs, denoted by OP, are formed as:

$$ColP = \{p_{-l} \ldots w_0 \ldots p_{+r} | l \geq 0; r \geq 0; l + r \geq 1; l + r \leq Len\}$$

$$OP = \{(p_i, i) | i = -l, \ldots, r\}$$

Syntactic Relations

[Hearst (1991)] segments text into noun phrases, prepositional phrases, and verb groups, and discards all other syntactic information. [Yarowsky (1993)] determines various behaviors based on syntactic category; for example, that verbs derive more disambiguating information from their objects than from their subjects, adjectives derive almost all disambiguating information from the nouns they modify, and nouns are best disambiguated by directly adjacent adjectives or nouns. In other works, syntactic information most often is simply part of speech, used invariably in conjunction with other kinds of information such as [Bruce & Wiebe (1994), Leacock (1998)]. Evidence suggests that different kinds of disambiguation procedures are needed depending on the syntactic category and other characteristics of the target word [Yarowsky (1993), Leacock (1998)]. Most recent studies also use syntactic information such as in [Lee & Ng(2002), Montoyo *et al.* (2005), Ando(2006)]. However, there is no an unique use of syntactic information through all these works. This circumstance can be seen, at least, in two aspects: the used syntactic parser, and the syntactic relations. For example, [Hearst (1991)] have avoided complex processing by using shallow parsing to achieve noun phrases, prepositional phrases, and verb groups, and then extract from these phrases the complementary components of the target word as the syntactic information; [Lee & Ng(2002)] parsed sentences containing the target word using a statistical parser in [Charniak (2000)], and then the generated constituent tree is converted into a dependency tree to obtain syntactic information; [Ando(2006)] used the Slot Grammar-based full parser ESG in [McCord (1990)] and extracted several syntactic relations such as subject-of, object-of, and noun modifier.

7.3 SMT

The noisy channel model is the basic model for phrase-based SMT [Koehn et al. (2003)]:

$$\arg\max_e P(e|f) = \arg\max_e [P(f|e) \times P(e)] \tag{7.3}$$

The model can be described as a generative story [1]. First, an English sentence e is generated with probability $P(e)$. Second, e is segmented into phrases $\overline{e}_1, ..., \overline{e}_I$ (assuming a uniform probability distribution over all possible segmentations). Third, e is reordered according to a distortion model. Finally, French phrases $\overline{f}_i$ are generated under a translation model $P(\overline{f}|\overline{e})$ estimated from the bilingual corpus. Though other phrase-based models follow a joint distribution model [Marcu and Wong (2002)], or use log-linear models [Och and Ney (2004)], the basic architecture of phrase segmentation, phrase reordering, and phrase translation remains the same.

[1] We follow the convention in [Brown et al. (1993)], designating the source language as "French" and the target language as "English".

Recently, conventional phrase-based SMT systems often use the following features:

- phrase translation (2)
- lexical translation (2)
- trigram language model
- target word penalty
- phrase penalty
- distortion

7.4 WSD for SMT

7.4.1 WSD Task

In order to use WSD for SMT, the precondition is training data. Manually-created data sets such as SENSEVAL and SemCor, which are often used in WSD studies, are too small for applications like machine translation. We overcome this difficulty by extracting training data from bilingual corpora, similarly to [Carpuat and Wu (2007)] and [Chan et al. (2007)]. Word alignment information serves as a map between source words and target words. Target words are seen as senses. Since word alignment is not perfect, the resulting WSD training data is noisy. When carrying out this research, we consider WSD for both word and phrase.

7.4.2 WSD Training Data Generation

A procedure for WSD-training-data extraction is:

- Input: a bilingual corpus, a POS-tagged version of the source text, word alignment information
- Output: WSD training sets for source phrases
- Step 1: Collect phrase pair instances associated with position in the bilingual corpus. Group phrase pairs according to source phrase.
- Step 2: For each group, generate a training set for its corresponding source phrase.

Phrase pairs (s,t) which are consistent with the word alignment will be generated. The criteria of consistence with word alignment [Koehn et al. (2003)] are as follows:

- There exist links from words of s to words of t
- For every word outside s, there is no link to any word of t

- For every word outside t, there is no link to any word of s

Two training examples for ”power” which is extracted from Europarl corpus, English-French language pair:

senseid=“**nergie**”

<context>

Biomass and hydroelectric <head>**power**</head> account for 95 % of renewable energy sources in the European Union .

</context>

senseid=“**pouvoir**”

<context>

Under the Amsterdam Treaty , the European Parliament does have the <head>**power**</head> of co-decision in the field of consumer protection and public health matters .

</context>

Now we observe what kind of target phrases (senses) can appear in the sense set of a source phrase. A part of ”power”’s sense set can be {l’ nergie, nergie, pouvoir, lectricit}. Here ”nergie” and ”pouvoir” are two major senses. ”l’ nergie” is a noun phrase in which ”l’” is a French definite article. ”lectricit” is resulted from incorrect word alignment because it often co-occurs with ”nergie”. A part of the sense set of ”powers” is: {comptences, comptence, pouvoirs, pouvoir, puissance}. There is no ”nergie” in this set because ”power” is an uncountable noun under this sense. The word ”powers” can be translated into a French word which is in either singular or plural form. Gender is also a source of lexical diversity, for example ”pouvoir” and ”puissance”. These words have the same meaning in certain contexts. However, ”pouvoir” has a masculine gender and ”puissance” feminine, so their usage is still different.

When extracting WSD training data from a bilingual corpus, the number of training sets resulting from the extractive procedure is often much larger than vocabulary size of the source text. As can be seen from the previous example, raw data extracted from a bilingual corpus is a miscellany of semantic, lexical, morphological, an syntactic ingredients. It is very different from conventional WSD data style. This data can be refined in some ways such as lemmatization. The effect of lemmatization on translation quality will be analyzed in Section 5.

7.4.3 WSD Features

In our work, we use six kinds of knowledge as mentioned above and represent them as subsets of features, as below:

- *bag-of-words*, $F_1(l, r) = \{w_{-l}, \ldots, w_{+r}\}$: We investigate three sets of this knowledge including $F_1^a = F_1(-5, +5)$, $F_1^b = F_1(-10, +10)$, $F_1^c = F_1(-100, +100)$, corresponding to small size, medium size, and large size, respectively.

- *collocation of words*, $F_2 = \{w_{-l} \ldots w_{+r}\}$: As a result of our work in [Le & Shimazu (2004)] we choose collocations such that their lengths (including the target word) are less or equal to 4, it means $(l + r + 1) \leq 4$.
- *ordered words*, $F_3 = \{w_i | i = -l, \ldots, +r\}$: We choose $l = r = 3$
- *collocation of POSs*, $F_4 = \{p_{-l} \ldots p_{+r}\}$: Like collocation of words, we choose their lengths including the target word are less or equal to 4.
- *ordered POSs*: $F_5 = \{p_i | i = -l, \ldots, +r\}$: We choose $l = r = 3$
- *syntactic relations*, $F_6 = \{(target\_word, relation\_type, pos\_tag, word)\}$: $relation\_type$ receives a value from set $\{subj, obj, head, mod\}$ where $subj$ (obj) denote subjective (objective) function of $target\_word$, $head$ (mod) represent modifier-head (head-modifier) relation between $target\_word$ and $word$. pos_tag and $word$ correspond to words which have a relation of type contained in $relation\_type$ with the target word denoted by $target\_word$.

In case we are working with a training set of a source phrase, features will be extracted from surrounding context of that phrase. Syntactic feature is an exception. The syntactic feature set of a phrase is computed through its words.

7.4.4 Integration

After having been trained, WSD models can be used as a feature for SMT. Since we use a log linear translation model, the use of a new feature is easy. Feature's weight is tuned using minimum error rate training (Och, 2003). In decoding phase, when translation options are generated, their WSD score is computed and then can be used in searching process. Among other features, this new feature is sensitive to large context.

Given a source phrase, the simplest way is to train its own WSD model and then apply that model in new contexts. The number of WSD models is equal to the number of source phrases in the SMT phrase table. An alternative is to score a phrase using shorter phrases. That means only WSD models for phrases whose length is smaller than a threshold to be trained. This setting could reduce computational time. Suppose that we are considering a phrase pair (s, t) in which s is a source phrase, t is a target phrase. If this phrase pair can be split into a sequence (s_i, t_i) of n sub phrase pairs which are consistent with the word alignment of (s, t), then the probability of t given s and its context can be computed using 7.4.

$$P_{wsd}(t|s) = \prod_{i=1}^{n} P_{wsd}(t_i|s_i) \tag{7.4}$$

$P_{wsd}(t_i|s_i)$ calculates the probability of t_i conditioning on s_i and its surrounding context. If there are more than one possible split, we use a greedy method. This method gives preferences to sub phrases according to their length and score.

[Carpuat and Wu (2007)] used only the scoring method that trained WSD models for every source phrases. [Chan et al. (2007)] proposed an algorithm to score synchronous CFG rules of a hierarchical phrase-based SMT system. In this study, we evaluate the effect of both full and short scoring methods for phrase-based SMT.

Table 7.1: Corpus statistics of English-Vietnamese translation task.

		English	Vietnamese
EV50001	Sentences	55,347	
	Average sentence length	11.26	10.63
	Words	622,965	588,554
	Vocabulary	23,936	24,245

Table 7.2: Corpus statistics of English-Japanese translation task.

		English	Japanese
Reuters	Sentences	56,778	
	Average sentence length	26.70	33.50
	Words	1,488,572	1,867,952
	Vocabulary	31,702	29,406

7.5 Experiments

7.5.1 Corpora and Tools

A number of tools used in our experiments can be listed as follows. Vietnamese sentences were segmented using a word-segmentation program [Nguyen et al. (2003)]. For learning phrase translations, we used Pharaoh [Koehn (2004)], a state-of-the-art phrase-based SMT system which is available for research purpose. For word alignment, we used the GIZA++ tool [Och and Ney (2000)]. For learning language models, we used SRILM toolkit [Stolcke (2002)]. For MT evaluation, we used BLEU measure [Papineni et al. (2001)] calculated by the NIST script version 11b. For morphological analysis, we used a rule-based morphological analyzer which is described in [Pham et al. (2003)].

We implemented a phrase-based decoder with features described in Section 3. This decoder uses techniques described in [Koehn (2004)]. Our decoder achieves the same performance as Pharaoh system [Koehn (2004)]. We used it for experiments in this section.

7.5.2 WSD Evaluation

A number of automatically-generated WSD data sets are chosen for WSD evaluation. We select only data sets whose size are greater than 100 and then remove senses (target phrases) which have smaller than 7 examples. Another restriction is that source phrases must contain at least one content word. Each data set is divided randomly into two parts, for testing and training respectively. Obviously, these data sets are noisy therefore experimental results will not be perfect. However, by carrying out this evaluation, at least we have an idea about how good WSD is on these sets. WSD accuracy is computed as in 7.5. Table 7.3 shows the performance of the MEM classifier on two corpora. On EV50001, the NB classifier achieved an accuracy of 0.65 (we do not show in the Table

Table 7.3: WSD accuracy of the MEM classifier.

Corpus	Test phrases	Occurrences	Senses	WSD accuracy
EV50001	513	69,951	1,669	0.753
Reuters	1096	249,484	4,652	0.671

7.3) which is much lower than that of the MEM classifier. The possible reason is that NB is not as good as MEM when data is noisy and sparse. In the remaining experiments, we employ the MEM classifier only.

$$accuracy = \frac{\text{the number of } \textit{correctly} \text{ detected labels predicted by the classifier}}{\text{the size of the test data}} \tag{7.5}$$

An automatically generated training set contains many translations which are interchangeable. This phenomenon is caused by synonym or paraphrase. The phenomenon leads to previous performance is actually lower than true disambiguation power of classifiers. The discrimination between those senses is not as important as between really different senses.

The problem is not only semantic disambiguation but also morphological and syntactic disambiguation. Source words and target words in inflected forms are mapped together (for example, a plural noun to a noun in the plural). Function word insertion and deletion also make phrase pairs diverse. Phrase itself is less semantically ambiguous than word. However, according to our observation, it is so often that a source phrase is mapped into several target phrases which are different in morphology and syntax. In that case the disambiguation evidence comes from syntactic context.

7.5.3 SMT Evaluation

We used a maximum phrase length of 7 for the decoder. Word alignment heuristic we used was growth-diagonal. Feature weights of the decoder are tuned using a MERT script [Koehn (2004)]. Column WSD-7 shows the best BLUE scores of the system with WSD feature. Using sign test, we verified that improvements were statistically significant. A large number of WSD models were used because for each source phrase, a separate WSD model was trained.

Table 7.5.3 shows the effect of WSD unit length on BLEU score. The higher unit length, the higher BLEU score. An interesting result is that a unit length of three is good enough. We do not have to train WSD models for all phrases but only for those whose length is up to three.

Corpus	Baseline	WSD-7	WSD-3	WSD-2	WSD-1
EV50001	36.57	37.50	37.46	37.15	36.71
Reuters	24.20	24.73	24.68		

Table 7.4: BLEU scores of the WSD-SMT system. MEM classifier is used. Since WSD-3 is very close to WSD-7, we do not need to compute WSD-4, WSD-5, and WSD-6.

Feature	all	POSs	words	*bag-of-words*	POSs+words
BLEU score	37.50	36.89	37.32	36.13	37.32

Table 7.5: BLEU scores with different WSD features. all=all kinds of features, POSs=*collocation of POSs* and *ordered POSs*, words=*collocation of words* and *ordered words*. MEM is used.

7.5.4 WSD Feature Evaluation

Table 7.5.4 shows translation performance with different WSD features. The best performance can be achieved with all kinds of features. If single POSs is used, the improvement is quite little. Words feature alone is the most effective. This result suggest that local context is very important for both WSD and machine translation. The combination of POSs and words is surprisingly no better than single words. The feature bag-of-words should be used in combination with other features because it can lessen the BLEU score if it is used separately.

Up to now, experiments do not involve syntactic relation features. We use a constituent parser [Charniak (2000)] to parse the source text of bilingual corpora. Then we extract word pairs according to head-modifier relation. This procedure will produce word pairs which have one of following syntactic relations: subject-verb, verb-object, adjective-head noun, noun modifier-head noun, preposition-noun, etc. We use this new feature for WSD MEM classifier. First, we carry out experiments using data sets described in Table 7.3. The WSD accuracy increase from 0.753 to 0.762 on EV50001 corpus and from 0.671 to 0.673 on Reuters corpus. Then we train new WSD models for SMT. On EV50001 corpus, the BLEU score increases slightly from 37.50 to 37.53. On Reuters corpus, the BLEU score decreases from 23.10 to 23.4, also not significantly.

7.6 Conclusions

We presented our empirical results of the WSD integration into SMT. We implemented the approach proposed by [Carpuat and Wu (2007)]. Our experiments reinformed that WSD can improve SMT significantly. We used two WSD models including MEM and NB while [Carpuat and Wu (2007)] used an ensemble of four combined WSD models (NB, MEM, Boosting, and Kernel PCA-based). Our experiments showed that the use of MEM is more effective than the use of NB. [Carpuat and Wu (2007)] trained WSD models for all phrases of length up to 7. Our experiments indicated that with only the length 3, the result is compared to 7. We presented a simple scoring method to accomplish that. We also conducted experiments employing syntactic relation features for WSD. However this feature did not bring a significant change to the performance of both WSD and SMT.

Chapter 8

Conclusions

We presented a number of tree-to-string phrase-based SMT approaches. The required resources to build such system include a source language parser, a word alignment tool, and a bilingual corpus. We proposed a syntactic transformation model based on the probabilistic context free grammar. We defined syntactic transformation including the word reordering, the deletion and the insertion of function words. This definition prevents our model from learning heavy grammars to solve the word choice problem. By using this model, we study several phrase-based SMT approaches:

- Phrase-based SMT with preprocessing: Source sentences are transformed in the preprocessing phase. We proposed a morphological transformation schema for English-Vietnamese translation. This approach can improve translation quality significantly.
- Phrase-based SMT with chunk-based reordering: This method can improve translation quality. Its main advantage is speed since shallow parsing is much faster than full parsing and decoding algorithm is also very fast.
- Syntax directed phrase-based SMT: This is a general frame word employing the syntactic transformation model in the decoding phase. This approach can also improve translation quality significantly.

We carried out an empirical study of WSD integration into SMT [Carpuat and Wu (2007), Chan et al. (2007)]. Our experiments reinformed that WSD can improve SMT significantly. We used two WSD models including MEM and NB while [Carpuat and Wu (2007)] used an ensemble model and [Chan et al. (2007)] used SVM. Our experiments indicated that directly training WSD models for phrases longer than 3 words does not have a strong impact on performance. We presented a simple phrase scoring method to accomplish that. We used syntactic relation features for WSD. However this feature did not make a significant change to the performance of SMT.

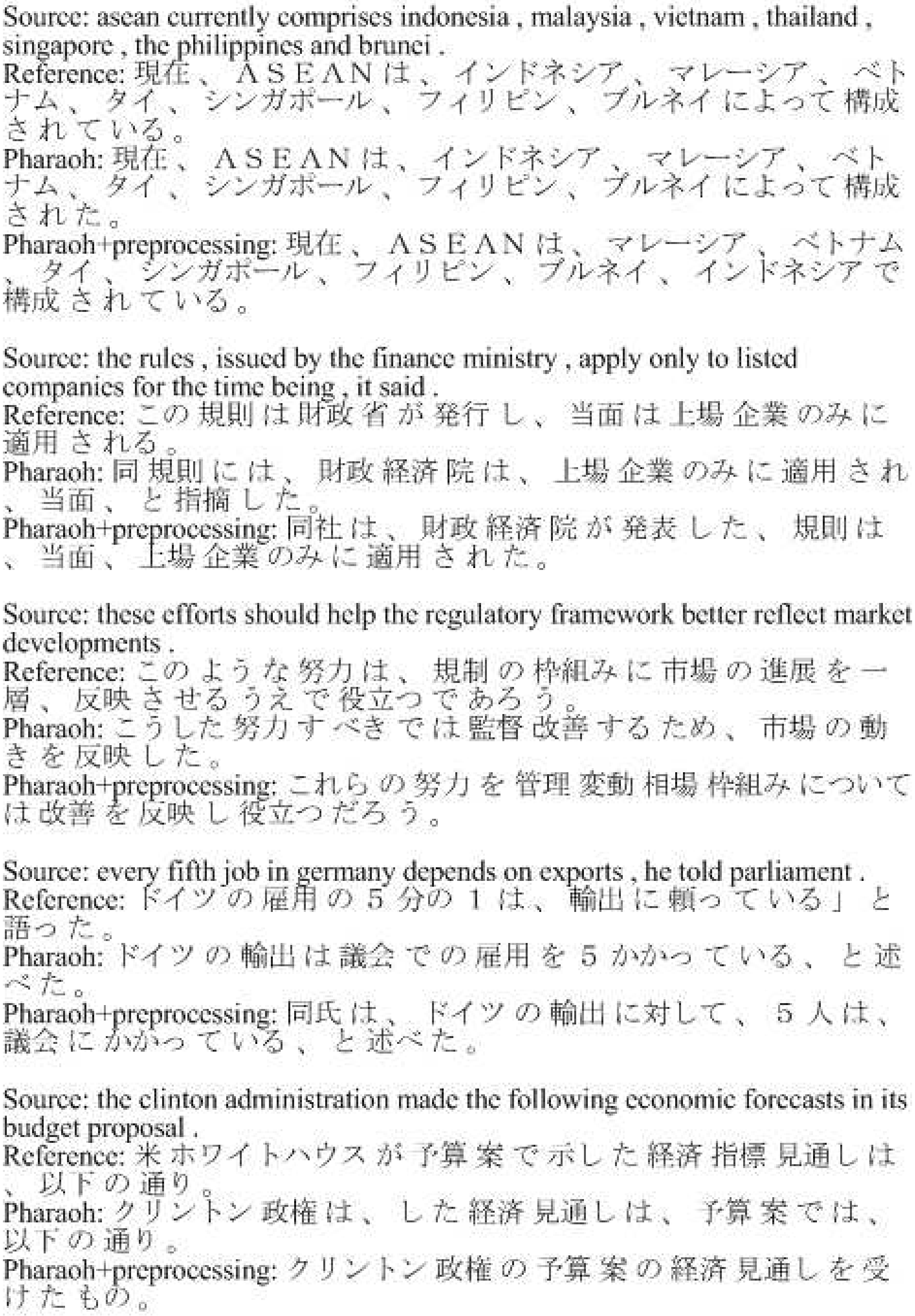
There are several ways to extend our frame work of syntax directed phrase-based SMT. First, syntactic parsing is not perfect especially when a parser trained on Penn Treebank comes to analyze texts in a different domain. Using a n-best list of parses instead of 1-best is an extension to improve translation quality. Our decoding algorithm in Chapter 6 should be upgraded to represent an input tree forest and to search over it. A second way to improve translation quality is to deal with the flexible of adjunct attachment. Our decoder should allow a movement of adjuncts without changes the dependency structure of the input syntactic tree. This treatment leads to deal with a set of parses whose dependency structure is the same.

We also intend to apply the syntactic transformation model to improve word alignment. The notable GIZA++ tool is an implementation of IBM translation models (Model 1, 2, 3, 4, and 5). All models are word-based. The input and the output of the noisy channel are just sequences of words. The channel's operations are word duplications (including insertion and deletion), word movements, and word translations. Using a string-to-tree noisy channel model for word alignment, we expect to improve word alignment accuracy for language pairs which are very different in word order such as English and Japanese.

Appendix

POS tag	Description	POS tag	Description
CC	Coordinating conjuction	TO	*to*
CD	Cardinal number	SYM	Symbol
DT	Determiner	UH	Interjection
EX	Existential *there*	VB	Verb, base form
FW	Foreign word	VBD	Verb, past tense
IN	Prep./subordinating conj	VBG	Verb, gerund/present participle
JJ	Adjective	VBN	Verb, past participle
JJR	Adjective, comparative	VBP	Verb, non-3rd singular present
JJS	Adjective, superlative	VBZ	Verb, 3rd singular present
LS	List item marker	WDT	Wh-determiner
MD	Modal	WP	Wh-pronoun
NN	Noun, singular/mass	WP$	Possessive wh-pronoun
NNS	Noun, plural	WRB	Wh-adverb
NNP	Proper noun, singular	,	Comma
NNPS	Proper noun, plural	.	Full stop
PDT	Predeterminer	“	Open quotation mark
POS	Possessive ending	”	Close quotation mark
PRP	Personal pronoun	:	Colon sign
PRP$	Possessive pronoun	$	Currency sign
RB	Adverb	(	Open parenthesis
RBR	Adverb, comparative	)	Close parenthesis
RBS	Adverb, superlative	#	Number sign
RP	Particle		

Table A.1: Penn Treebank’s part-of-speech tags



Source: asean currently comprises indonesia , malaysia , vietnam , thailand , singapore , the philippines and brunei .
Reference: 現在 、 ＡＳＥＡＮ は 、 インドネシア 、 マレーシア 、 ベトナム 、 タイ 、 シンガポール 、 フィリピン 、 ブルネイ によって 構成 さ れ て いる 。
Pharaoh: 現在 、 ＡＳＥＡＮ は 、 インドネシア 、 マレーシア 、 ベトナム 、 タイ 、 シンガポール 、 フィリピン 、 ブルネイ によって 構成 さ れ た 。
Pharaoh+preprocessing: 現在 、 ＡＳＥＡＮ は 、 マレーシア 、 ベトナム 、 タイ 、 シンガポール 、 フィリピン 、 ブルネイ 、 インドネシア で 構成 さ れ て いる 。

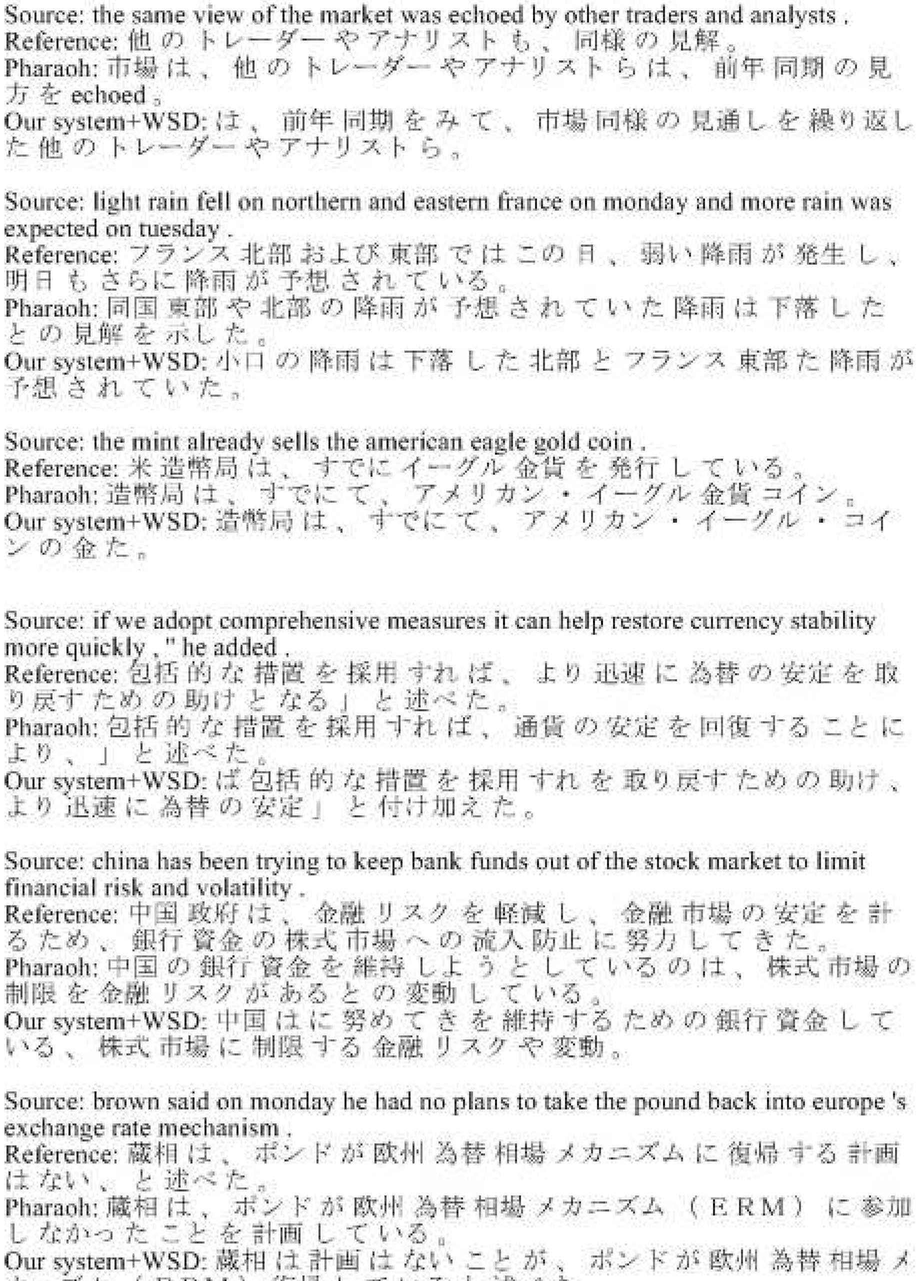
Source: the rules , issued by the finance ministry , apply only to listed companies for the time being , it said .
Reference: この 規則 は 財政 省 が 発行 し 、 当面 は 上場 企業 のみ に 適用 さ れる 。
Pharaoh: 同 規則 に は 、 財政 経済 院 は 、 上場 企業 のみ に 適用 さ れ 、 当面 、 と 指摘 し た 。
Pharaoh+preprocessing: 同社 は 、 財政 経済 院 が 発表 し た 、 規則 は 、 当面 、 上場 企業 のみ に 適用 さ れ た 。

Source: these efforts should help the regulatory framework better reflect market developments .
Reference: この よう な 努力 は 、 規制 の 枠組み に 市場 の 進展 を 一層 、 反映 さ せる うえ で 役立つ で あろ う 。
Pharaoh: こう し た 努力 す べき で は 監督 改善 する ため 、 市場 の 動き を 反映 し た 。
Pharaoh+preprocessing: これら の 努力 を 管理 変動 相場 枠組み について は 改善 を 反映 し 役立つ だろ う 。

Source: every fifth job in germany depends on exports , he told parliament .
Reference: ドイツ の 雇用 の ５ 分の １ は 、 輸出 に 頼っ て いる 」 と 語っ た 。
Pharaoh: ドイツ の 輸出 は 議会 で の 雇用 を ５ かかっ て いる 、 と 述べ た 。
Pharaoh+preprocessing: 同氏 は 、 ドイツ の 輸出 に対して 、 ５ 人 は 、 議会 に かかっ て いる 、 と 述べ た 。

Source: the clinton administration made the following economic forecasts in its budget proposal .
Reference: 米 ホワイトハウス が 予算 案 で 示し た 経済 指標 見通し は 、 以下 の 通り 。
Pharaoh: クリントン 政権 は 、 し た 経済 見通し は 、 予算 案 で は 、 以下 の 通り 。
Pharaoh+preprocessing: クリントン 政権 の 予算 案 の 経済 見通し を 受け た もの 。

Figure A.1: Examples of English-Japanese translation with preprocessing on Reuters corpus.



Source: the same view of the market was echoed by other traders and analysts .
Reference: 他 の トレーダー や アナリスト も 、 同様 の 見解 。
Pharaoh: 市場 は 、 他 の トレーダー や アナリスト ら は 、 前年 同期 の 見方 を echoed 。
Our system+WSD: は 、 前年 同期 を み て 、 市場 同様 の 見通し を 繰り返し た 他 の トレーダー や アナリスト ら 。

Source: light rain fell on northern and eastern france on monday and more rain was expected on tuesday .
Reference: フランス 北部 および 東部 で は この 日 、 弱い 降雨 が 発生 し 、 明日 も さらに 降雨 が 予想 さ れ て いる 。
Pharaoh: 同国 東部 や 北部 の 降雨 が 予想 さ れ て い た 降雨 は 下落 し た と の 見解 を 示し た 。
Our system+WSD: 小口 の 降雨 は 下落 し た 北部 と フランス 東部 た 降雨 が 予想 さ れ て い た 。

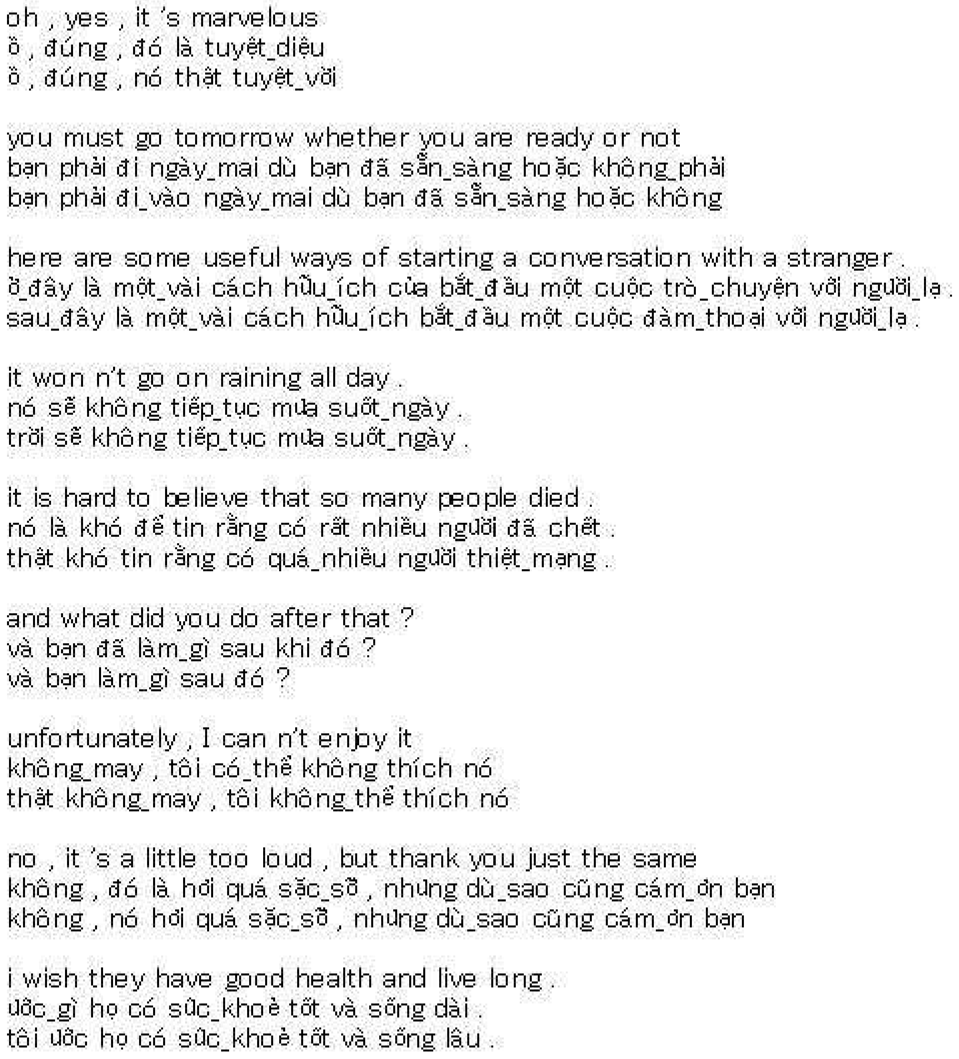
Source: the mint already sells the american eagle gold coin .
Reference: 米 造幣局 は 、 すでに イーグル 金貨 を 発行 し て いる 。
Pharaoh: 造幣局 は 、 すでに て 、 アメリカン ・ イーグル 金貨 コイン 。
Our system+WSD: 造幣局 は 、 すでに て 、 アメリカン ・ イーグル ・ コイン の 金 た 。

Source: if we adopt comprehensive measures it can help restore currency stability more quickly , " he added .
Reference: 包括 的 な 措置 を 採用 すれ ば 、 より 迅速 に 為替 の 安定 を 取り戻す ため の 助け と なる 」 と 述べ た 。
Pharaoh: 包括 的 な 措置 を 採用 すれ ば 、 通貨 の 安定 を 回復 する こと に より 、 」 と 述べ た 。
Our system+WSD: ば 包括 的 な 措置 を 採用 すれ を 取り戻す ため の 助け 、 より 迅速 に 為替 の 安定 」 と 付け加え た 。

Source: china has been trying to keep bank funds out of the stock market to limit financial risk and volatility .
Reference: 中国 政府 は 、 金融 リスク を 軽減 し 、 金融 市場 の 安定 を 計る ため 、 銀行 資金 の 株式 市場 へ の 流入 防止 に 努力 し て き た 。
Pharaoh: 中国 の 銀行 資金 を 維持 しよう と し て いる の は 、 株式 市場 の 制限 を 金融 リスク が ある と の 変動 し て いる 。
Our system+WSD: 中国 は に 努め て き を 維持 する ため の 銀行 資金 し て いる 、 株式 市場 に 制限 する 金融 リスク や 変動 。

Source: brown said on monday he had no plans to take the pound back into europe 's exchange rate mechanism .
Reference: 蔵相 は 、 ポンド が 欧州 為替 相場 メカニズム に 復帰 する 計画 は ない 、 と 述べ た 。
Pharaoh: 蔵相 は 、 ポンド が 欧州 為替 相場 メカニズム （ ＥＲＭ ） に 参加 し なかっ た こと を 計画 し て いる 。
Our system+WSD: 蔵相 は 計画 は ない こと が 、 ポンド が 欧州 為替 相場 メカニズム （ ＥＲＭ ） 復帰 し て いる と 述べ た 。

Figure A.2: Examples of English-Japanese translation with WSD integration on Reuters corpus.



oh , yes , it 's marvelous
ồ , đúng , đó là tuyệt_diệu
ồ , đúng , nó thật tuyệt_vời

you must go tomorrow whether you are ready or not
bạn phải đi ngày_mai dù bạn đã sẵn_sàng hoặc không_phải
bạn phải đi_vào ngày_mai dù bạn đã sẵn_sàng hoặc không

here are some useful ways of starting a conversation with a stranger .
ở_đây là một_vài cách hữu_ích của bắt_đầu một cuộc trò_chuyện với người_lạ .
sau_đây là một_vài cách hữu_ích bắt_đầu một cuộc đàm_thoại với người_lạ .

it won n't go on raining all day .
nó sẽ không tiếp_tục mưa suốt_ngày .
trời sẽ không tiếp_tục mưa suốt_ngày .

it is hard to believe that so many people died .
nó là khó để tin rằng có rất nhiều người đã chết .
thật khó tin rằng có quá_nhiều người thiệt_mạng .

and what did you do after that ?
và bạn đã làm_gì sau khi đó ?
và bạn làm_gì sau đó ?

unfortunately , I can n't enjoy it
không_may , tôi có_thể không thích nó
thật không_may , tôi không_thể thích nó

no , it 's a little too loud , but thank you just the same
không , đó là hơi quá sặc_sỡ , nhưng dù_sao cũng cám_ơn bạn
không , nó hơi quá sặc_sỡ , nhưng dù_sao cũng cám_ơn bạn

i wish they have good health and live long .
ước_gì họ có sức_khoẻ tốt và sống dài .
tôi ước họ có sức_khoẻ tốt và sống lâu .

Figure A.3: Examples of English-Vietnamese translation with WSD integration on EV50001 corpus. For each example, the first sentence is a source sentence, the second is the output of our phrase-based system, the third is the output of our system with WSD integration.

Bibliography

[Agirre & Martinez (2001)] E. Agirre, and D. Martinez, 2001a. Decision lists for english and basque. *Proceedings of the SENSEVAL-2 Workshop. In conjunction with ACL/EACL* Toulouse, France.

[Aho & Ullman (1972)] Aho, A. V., and Jeffrey D. Ullman, 1972. The Theory of Parsing, Translation, and Compiling, volume I: Parsing. Prentice Hall, Englewood Cliffs, New Jersey.

[Al-Onaizan et al. (1999)] Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky, 1999. Statistical machine translation. Final Report, JHU Summer Workshop.

[Ando(2006)] R. K. Ando, 2006. Applying Alternating Structure Optimization to Word Sense Disambiguation, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, pages 77–84.

[Bikel (2004)] Bikel, D. M., 2004. Intricacies of Collins' Parsing Model. *Computational Linguistics*, 30(4): 479-511.

[Brown et al. (1993)] Brown, P. F., S. A. D. Pietra, V. J. D. Pietra, R. L. Mercer, 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 22(1): 39-69.

[Bruce & Wiebe (1994)] R. Bruce and J. Wiebe, 1994. Word Sense Disambiguation using Decomposable Models. *Proceedings of ACL*, pages 139–145.

[Cabezas and Resnik (2005)] Clara Cabezas and Philip Resnik, 2005. Using WSD techniques for lexical selection in statistical machine translation. Technical report, Institute for Advanced Computer Studies, University of Maryland.

[Callison-Burch et al. (2006)] Callison-Burch, C., Miles Osborne, and Philipp Koehn, 2006. Re-evaluating the Role of Bleu in Machine Translation Research. In *Proceedings of EACL.*

[Carpuat and Wu (2005)] Marine Carpuat and Dekai Wu, 2005. Word Sense Disambiguation vs. Statistical Machine Translation. *Proceedings of ACL*, pages 387–394.

[Carpuat and Wu (2006)] Marine Carpuat, Y. Shen, X. Yu, and Dekai Wu, 2006. Toward Integrating Word Sense and Entity Disambiguation into Statistical Machine Translation. *Proceedings of IWSLT*, pages 37–44.

[Carpuat and Wu (2007)] Marine Carpuat and Dekai Wu, 2007. Improving Statistical Machine Translation Using Word Sense Disambiguation. *Proceedings of EMNLP-CoNLL.*

[Chan et al. (2007)] Y. S. Chan, H. T. Ng, and D. Chiang, 2007. Word Sense Disambiguation Improves Statistical Machine Translation. *Proceedings of ACL.*

[Charniak (2000)] Charniak, E., 2000. A maximum entropy inspired parser. In *Proceedings of HLT-NAACL.*

[Charniak et al. (2003)] Charniak, E., K. Knight, and K. Yamada, 2003. Syntax-based language models for statistical machine translation. In *Proceedings of the MT Summit IX.*

[Chiang (2005)] David Chiang, 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL.*

[Collins (1999)] Collins, M., 1999. *Head-Driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania.

[Collins et al. (2005)] Collins, M., P. Koehn, and I. Kucerova, 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL.*

[Cook (1988)] Cook, V. J., 1988. Chomsky's Universal Grammar: An Introduction. Basil Blackwell.

[Costa-jussia et al. (2007)] Costa-jussia, M. R., J. M. Crego, P. Lambert, M. Khalilov, J. A. R. Fonollosa, J. B. Marino, and R. E. Banchs, 2007. Ngram-based statistical machine translation enhanced with multiple weighted reordering hypotheses. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 167-170.

[Ding and Palmer (2005)] Ding, Y. and M. Palmer, 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of ACL.*

[Dung (2003)] Dung, V., 2003. Tieng Viet va ngon ngu hoc hien dai so khao ve cu phap. *VIET Stuttgart, Germany.*

[Escudero *et al.* (2000a)] G. Escudero, L. Marquez, and G. Rigau, 2000. Naive Bayes and exemplar-based approaches to Word Sense Disambiguation revisited. *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, pages 421–425.

[Escudero *et al.* (2000b)] G. Escudero, L. Marquez, and G. Rigau, 2000. Boosting Applied to Word Sense Disambiguation. *Proceedings of the 11th European Conference on Machine Learning (ECML)*, pages 129–141.

[Fox (2002)] Fox, H., 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of EMNLP.*

[Gale *et al.* (1993)] Gale, William A., Kenneth W. Church, and David Yarowsky. 1993. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

[Goldwater & McClosky (2005)] Goldwater, S. and D. McClosky, 2005. Improving statistical MT through morphological analysis. In *Proceedings of EMNLP*.

[Gunji (1987)] Gunji, T., 1987. Japanese Phrase Structure Grammar. *D. Reidel Publishing Company.*

[Hearst (1991)] M.A.Hearst, 1991. Noun homograph disambiguation using local context in large corpora. *Proceedings of the Seventh Annual Conference of the Centre for the New OED and Text Research: Using Corpora*, pages 1–22, Oxford, UK.

[Ide *et al.* (1998)] N. Ide and J. Véronis, 1998. Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics* Vol. 24, pages 1–40.

[Johnson (2002)] Johnson, M., 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of ACL*.

[Klein and Manning (2003)] Klein, D. and C. D. Manning, 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*.

[Knight (1999)] Knight, K., 1999. Decoding Complexity in Word-Replacement Translation Models. Computational Linguistics, Squibs & Discussion, 25(4).

[Knight and Graehl (2005)] Knight, K. and J. Graehl, 2005. An overview of probabilistic tree transducers for natural language processing. In *Proceedings of CICLing*.

[Koehn et al. (2003)] Koehn, P., F. J. Och, and D. Marcu, 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL*.

[Koehn (2004)] Koehn, P. , 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA*.

[Koehn & Hoang (2007)] Philipp Koehn and Hieu Hoang, 2007. Factored Translation Models. In *Proceedings of EMNLP*.

[Kuno (1981)] Kuno, S., 1981. The Structure of the Japanese Language. *MIT Press*.

[Le & Shimazu (2004)] C.A. Le and A. Shimazu, 2004. High Word Sense Disambiguation Using Naive Bayesian Classifier with Rich Features. *The 18th Pacific Asian Conference on Linguistic Information and Computation (PACLIC18)*, pages 105–113.

[Le *et al.* (2005a)] C.A. Le, V.N. Huynh, H.C Dam, A. Shimazu, 2005. Combining Classifiers Based on OWA Operators with an Application to Word Sense Disambiguation. *Proceedings of RSFDGrC*, Vol. 1, pages 512–521.

[Leacock (1998)] C. Leacock, M. Chodorow, and G. Miller, 1998. Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, pages 147–165.

[Lee (2004)] Lee, Y., 2004. Morphological analysis for statistical machine translation. In *Proceedings of NAACL*.

[Lee & Ng(2002)] Y. K. Lee and H. T. Ng, 2002. An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. *Proceedings of EMNLP*, pages 41–48.

[Lehmann (1986)] Lehmann, E. L., 1986. Testing Statistical Hypotheses (Second Edition). Springer-Verlag.

[Liu (2006)] Liu, Y., Qun Liu, Shouxun Lin, 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of ACL*.

[Marcu and Wong (2002)] Marcu, D. and W. Wong, 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP*.

[Marcu et al. (2006)] Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight, 2006. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. In *Proceedings of EMNLP*.

[Marcus et al. (1993)] Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz, 1993. Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics*, 19: 313-330.

[McCord (1990)] M. C. McCord, 1990. Slot Grammar: A system for simpler construction of practical natural language grammars. *Natural Language and Logic: International Scientific Symposium*, Lecture Notes in Computer Science, pages 118–145.

[Melamed (2004)] Melamed, I. D., 2004. Statistical machine translation by parsing. In *Proceedings of ACL*.

[Montoyo *et al.* (2005)] A. Montoyo, A. Suarez, G. Rigau and M. Palomar, 2005. Combining knowledge and corpus-based Word-Sense-Disambiguation methods. *Journal of Artificial Intelligence Research*, 23: 299–330.

[Mooney (1996)] R.J. Mooney, 1996. Comparative Experiments on Disambiguating Word Senses: An Illustration of The Role of Bias in Machine Learning. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–91.

[Ng & Lee (1996)] H.T. Ng and H.B. Lee, 1996. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. *Proceedings of ACL*, pages 40–47.

[Ng (1997)] H. Ng, 1997. Exemplar-Based Word Sense Disambiguation: Some Recent Improvements. *Proceedings of EMNLP*.

[Ngai *et al.* (2004)] G. Ngai, D. Wu, M. Carpuat, C-S. Wang, and C-Y. Wang, 2004. Semantic Role Labeling with Boosting, SVMs, Maximum Entropy, SNOW, and Decision Lists. *Proceedings of Workshop on Senseval-3*, Barcelona.

[Nguyen and Shimazu (2006a)] Nguyen, T. P. and Akira Shimazu, 2006. Improving Phrase-Based SMT with Morpho-Syntactic Analysis and Transformation. In *Proceedings of AMTA*.

[Nguyen and Shimazu (2006b)] Nguyen, T. P. and Akira Shimazu, 2006. Improving Phrase-Based Statistical Machine Translation with Morphosyntactic Transformation. *Machine Translation*, Vol. 20, No. 3, pp 147-166.

[Nguyen et al. (2007)] Nguyen, T. P., Akira Shimazu, Le-Minh Nguyen, and Van-Vinh Nguyen, 2007. A Syntactic Transformation Model for Statistical Machine Translation. *International Journal of Computer Processing of Oriental Languages (IJCPOL)*, Vol. 20, No. 2, 1-21.

[Nguyen et al. (2003)] Nguyen, T. P., Nguyen V. V. and Le A. C. Vietnamese Word Segmentation Using Hidden Markov Model. In *Proceedings of International Workshop for Computer, Information, and Communication Technologies in Korea and Vietnam*, 2003.

[Niessen & Ney (2004)] Niessen, S. and H. Ney, 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2):181-204.

[Och and Ney (2000)] Och, F. J. and H. Ney, 2000. Improved statistical alignment models. In *Proceedings of ACL*.

[Och and Ney (2004)] Och, F. J. and H. Ney, 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417-449.

[Och et al. (2004)] Och, F. J., D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of HLT-NAACL*.

[Papineni et al. (2001)] Papineni, K., S. Roukos, T. Ward, W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Report, 2001.

[Pedersen (2000)] T. Pedersen, 2000. A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. *Proceedings of NAACL*, pages 63–69.

[Petrov et al. (2006)] Petrov, S., Leon Barrett, Romain Thibaux, Dan Klein, 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of ACL*

[Pham et al. (2003)] Pham, N. H., Nguyen L. M., Le A. C., Nguyen P. T., and Nguyen V. V. LVT: An English-Vietnamese Machine Translation System. In *Proceedings of FAIR*, 2003.

[Quirk et al. (2005)] Quirk, C., A. Menezes, and C. Cherry, 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of ACL*.

[Sha and Pereira (2003)] F. Sha and F. Pereira, 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL 2003*.

[Shen et al. (2004)] Shen, L., A. Sarkar, F. J. Och, 2004. Discriminative reranking for machine translation. In *Proceedings of HLT-NAACL*.

[Stolcke (2002)] Stolcke, A. SRILM - An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002.

[Towell & Voorhees (1998)] Towell, G. G., & Voorhees, E. M. (1998). Disambiguating highly ambiguous words. *Computational Linguistics*, 24 (1), 125–145.

[Utiyama & Isahara (2003)] Utiyama, M., and Hitoshi Isahara, 2003. Reliable Measures for Aligning Japanese-English News Articles and Sentences. In *Proceedings of ACL*, pp. 72–79.

[Varea et al. (2001)] Varea, I. G., F. J. Och, H. Ney, and F. Casacuberta, 2001. Refined Lexicon Models for Statistical Machine Translation using a Maximum Entropy Approach. *Proceedings of ACL*, pages 204–211.

[Xia and McCord (2004)] Xia, F. and M. McCord, 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of COLING.*

[Yamada and Knight (2001)] Yamada, K. and K. Knight. A syntax-based statistical translation model, 2001. In *Proceedings of ACL.*

[Yarowsky (1992)] D. Yarowsky, 1992. Word Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *Proceedings of COLING*, pages 454–460.

[Yarowsky (1993)] Yarowsky, David. 1993. One sense per collocation. *Proceedings of ARPA Human Language Technology Workshop*, pages 266–271, Princeton, NJ.

[Yarowsky (1994)] Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. *Proceedings ACL*

[Zhang et al. (2007)] Zhang, Y., Richard Zens, and Hermann Ney, 2007. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In *Proceedings of the NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation.*

[Zollmann and Venugopal (2006)] Zollmann, A., and Ashish Venugopal, 2006. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings of the SMT Workshop, HLT-NAACL.*

Publications

Journals

[1] Nguyen Phuong Thai, Akira Shimazu, 2006. Improving Phrase-Based Statistical Machine Translation with Morphosyntactic Transformation. *Machine Translation*, Vol. 20, No. 3, pp 147-166.

[2] Nguyen Phuong Thai, Akira Shimazu, Le-Minh Nguyen, and Van-Vinh Nguyen, 2007. A Syntactic Transformation Model for Statistical Machine Translation. *International Journal of Computer Processing of Oriental Languages (IJCPOL)*, Vol. 20, No. 2, 1-21.

Referred International Conferences

[3] Nguyen Phuong Thai, Akira Shimazu. Improving Phrase-Base SMT with Morpho-Syntactic Analysis and Transformation. *The 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 06).* August 8-12, 2006, Cambridge, U.S.

[4] Nguyen Phuong Thai, Akira Shimazu. A Syntactic Transformation Model for Statistical Machine Translation. *The 21st International Conference on Computer Processing of Oriental Languages (ICCPOL 06).* December 17-19, 2006, Singapore.

[5] Nguyen Phuong Thai, Akira Shimazu. Rule-Based Transformation for Improving Phrase-Base SMT from English to Vietnamese. *The 1st International Conference on Knowledge, Information and Creativity Support Systems (KICSS 06).* August 1-4, 2006, Thailand.

[6] Le-Minh Nguyen, Akira Shimazu, Nguyen Phuong Thai, Xuan-Hieu Phan. A Multilingual Dependency Analysis System Using Online Passive-Aggressive Learning, *Shared Task paper Proceedings of EMNLP-CONLL 2007*, pp 1149-1155.