Improving Phrase-Based SMT with Morpho-Syntactic Analysis and Transformation

Thai Phuong Nguyen and **Akira Shimazu**
School of Information Science
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{thai,shimazu}@jaist.ac.jp

Abstract

This paper presents our study of exploiting morpho-syntactic information for phrase-based statistical machine translation (SMT). For morphological transformation, we use hand-crafted transformational rules. For syntactic transformation, we propose a transformational model based on Bayes' formula. The model is trained using a bilingual corpus and a broad coverage parser of the source language. The morphological and syntactic transformations are used in the preprocessing phase of a SMT system. This preprocessing method is applicable to language pairs in which the target language is poor in resources. We applied the proposed method to translation from English to Vietnamese. Our experiments showed a BLEU-score improvement of more than 3.28% in comparison with Pharaoh, a state-of-the-art phrase-based SMT system.

1 Introduction

In the field of statistical machine translation (SMT), several phrase-based SMT models (Och et al., 1999; Marcu and Wong, 2002; Koehn et al., 2003) have achieved the 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 modeling 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 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 paper, we describe our study for improving SMT by using linguistic analysis to attack these two problems. 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 are closer in word-order to the target language than the original sentences. We propose a transformational model based on the Bayes formula. The model's knowledge is learned from bitext in which the source text has been parsed. The sparse data problem is solved by splitting the stem and the inflectional suffix of a word during translation. These

Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pages 138-147,
Cambridge, August 2006.

Table 1. Preprocessing procedure

+ 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

morpho-syntactic transformations are applied in the preprocessing phase of a SMT system (Figure 1). The preprocessing procedure takes in a source sentence and performs five steps as shown in Table 1. This preprocessing procedure was applied to source sentences in both the training and testing phases.

The rest of this paper is organized as follows: In Section 2, background information is presented. Section 3 describes the syntactic transformation. Section 4 presents the morphological transformation. Finally, Section 5 discusses our experimental results.

2 Background

2.1 Phrase-Based SMT

The noisy channel model is the basic model of the phrase-based SMT (Koehn et al., 2003):

$$\arg\max_{e} P(e \mid f) = \arg\max_{e} P(f \mid e) \times P(e)$$

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 $\bar{e}_1,...,\bar{e}_I$ (assuming a uniform probability distribution over all possible segmentations). Third, *e* is reordered according to some distortion model. Finally, French phrases $\bar{f}_i$ are generated under a translation model $P(\bar{f} \mid \bar{e})$ 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.

2.2 Approaches to Exploit Syntactic Information for SMT

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

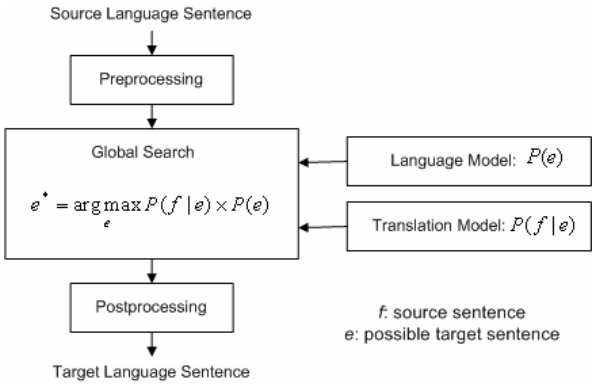

Figure 1. Architecture of a SMT system

Several previous studies have proposed translation models which incorporate syntactic representations of the source and/or target languages. Yamada and Knight (2001) proposed a new SMT model that uses syntax information in the source language alone. The model is based on a tree-to-string noisy channel model, and the translation task is transformed into a parsing problem. Melamed (2004) used synchronous context-free grammars (CFGs) for parsing both languages simultaneously. This study showed that syntax-based SMT systems could be built using synchronous parsers.

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.

A third 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. Our study differs from those of (Xia and McCord, 2004) and (Collins et al., 2005) in several impor-

tant respects. First, our transformational model is based on statistical decisions, 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 employed parsers of both source and target languages. Third, we consider translation from English to Vietnamese. Last, we use syntactic transformation in combination with morphological transformation.

Reranking (Koehn et al., 2003; Shen 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).

2.3 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 and Ney (2000) represented 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 and McClosky (2005) recently used the techniques proposed in (Al-Onaizan et al., 1999) for Czech-English language pair, with some refinements, and analyzed the usefulness of each morphological 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.

2.4 Vietnamese Language Features

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. For example[2]:

> 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), .

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 to a Vietnamese function word which precedes and modifies the head word (for example, "book*s*" → "book–*s*" → "*những* cuốn sách", or "work*ing*" → "work-*ing*" → "*đang* làm việc"). English derivative words often correspond to Vietnamese compound words (for example, "change*ably*" → "thay_đổi *được*").

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

The third difference is word order. Vietnamese has a SVO sentence form[3] similar to English. For example:

English sentence: I see him .

Vietnamese sentence: Tôi nhìn anh_ấy .

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. For example:

English sentence: Who does he love ?

Vietnamese sentence: Anh_ấy yêu ai ?

In addition, most Vietnamese yes/no questions end with an interrogative word, while the English yes/no questions do not. For example:

English sentence: Do you love her ?

Vietnamese sentence: Anh yêu cô_ấy phải_không ?

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 premodifiers to follow the head word. For example:

Original English noun phrase: his friend 's book

Vietnamese phrase: quyển_sách của bạn anh_ấy

Transformed English noun phrase: book 's friend his

Vietnamese phrase: quyển_sách của bạn anh_ấy

3 Syntactic Transformation

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[4] NP → DT JJ NN in English can become NP → DT NN JJ or NP → NN JJ DT in Vietnamese. For the phrase "a nice weather", the first reordering is most appropriate, while for the phrase "this nice weather", 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.

3.1 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 $LHS_i \rightarrow RHS_i$, $1 \le i \le n$, (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 *(LHS → RHS, RS)* which is a pair consisting of an unlexicalized CFG rule and a reordering sequence (RS). For example, the rule (NP → JJ NN, 1 0) implies that the CFG rule NP → JJ NN in source language can be transformed into the rule NP → 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. Using the Bayes formula, we have:

$$P(T \mid S) = \frac{P(S \mid T) \times P(T)}{P(S)}$$

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:

$$Q^* = \arg\max_{Q}[P(S \mid T) \times P(T)] \quad (1)$$

Each transformational rule has an associated probability. Transformational rules whose CFG rules are the same form a group. A group is ambiguous if it contains more than one element. The probabilities of transformational rules in a group must sum to 1. P(S|T) is then decomposed into[5]:

$$P(S \mid T) = \prod_{i=1}^{n} P(LHS_i \rightarrow RHS_i, RS_i)$$

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

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

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

[5] For simplicity, we use $LHS_i \rightarrow RHS_i$ to represent the unlexicalized CFG rule in a transformational rule.

$$P(T) = \prod_{i=1}^{n} P(LHS_i \rightarrow RHS_i')$$

where $LHS_i \rightarrow RHS_i'$ is the result of reordering $LHS_i \rightarrow RHS_i$ using RS_i.

Since application of a transformational rule only reorders the right-hand-side symbols of a CFG rule, we can rewrite (1):

$$Q^* = \{RS_i^* : RS_i^* = \underset{RS_i}{\operatorname{argmax}}[P(LHS_i \rightarrow RHS_i, RS_i) \times P(LHS_i \rightarrow RHS_i')], \quad i = 1, \ldots, n\} \tag{2}$$

Suppose that a lexicalized CFG rule has the following form:

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

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*, and *k* and *m* are *0* in unary rules. Since the number of possible lexicalized rules is huge, direct estimation of *P(LHS → RHS)* is not feasible. Therefore the rule-markovization technique (Collins, 1999; Charniak, 2000; Klein and Manning, 2003) can be applied here. Given the left hand side, the generation process of the right hand side can be decomposed into three steps:

- Generate the head constituent label with probability $P_H = P(H \mid F, h)$
- Generate the right modifiers with probability:

  $$P_R = \prod_{i=1}^{k+1} P(R_i(r_i) \mid F, h, H)$$

  where $R_{k+1}(r_{k+1})$ is a *STOP* symbol which is added to the set of nonterminal symbols. The grammar model stops generating right modifiers when *STOP* is generated.
- Generate the left modifiers with probability:

  $$P_L = \prod_{i=1}^{m+1} P(L_i(l_i) \mid F, h, H)$$

  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 modifiers). Higher orders can be used if necessary. The probability of a lexicalized CFG rule now becomes:

$$P(LHS \rightarrow RHS) = P_H \times P_R \times P_L$$

In our experiments, we implemented Collins' Grammar Model 1 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 transformation of source language trees. In the next section, we will show how we induced this kind of data.

3.2 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 transformed syntactic tree are induced. Then the probabilities of transformational rules are computed. Finally, the transformed syntactic trees are used to train the LPCFG.

Figure 2 shows an example of inducing transformational rules. Source sentence and target sentence are in the middle part of the figure, on the left. The source syntactic tree is in the upper left part of the figure. Word links are represented by dotted lines. Words and aligned phrases of the target sentence are represented by lines (in the left lower part of the figure). Word alignment result, hierarchical alignment result, and induced transformational rules are in the lower right part of the figure. The transformed tree is in the upper right.

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):

$$score(s,t) = \frac{links(s,t)}{words(s) + words(t)} \tag{3}$$

where s is a source phrase, t is a target phrase; links(s,t) is the total number of source words in s

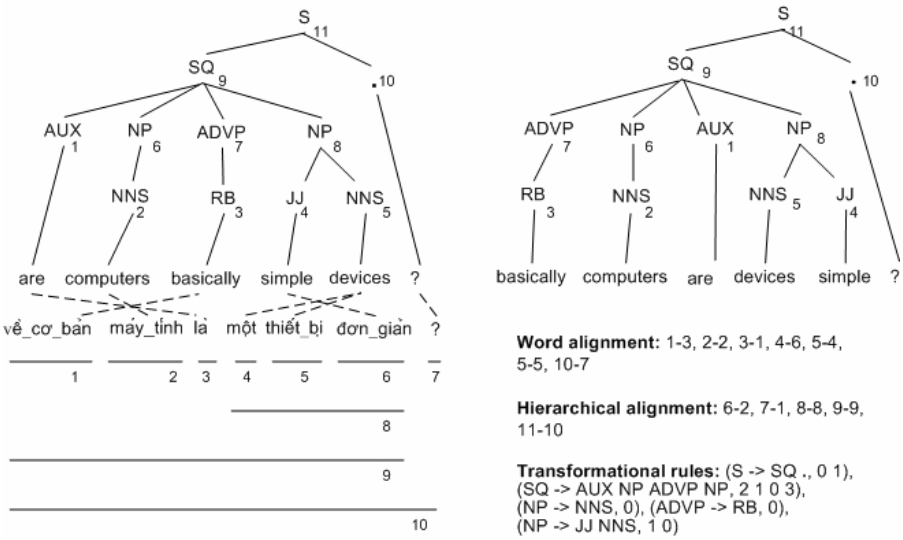

Figure 2. Inducing transformational rules

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 chose the best link based on the intersection of directional alignments, and on word link score. When applying formula (3) in determining alignment of a source constituent, if there were several target phrases having the highest link score, we used additional criteria:

- 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

Given a hierarchical alignment, a transformational rule can be computed for each constituent of the source syntactic tree. For a source constituent X with children X_1, ..., X_n and their aligned target phrases Y, Y_1, ..., Y_n, the conditions for inducing the transformational rule are as follows:

- Y_i are adjacent to each other
- Y contains Y_1, ..., Y_n but not any other target phrases

Suppose that f is a function in which $f(j)=i$ if X_i is aligned to Y_j. If the conditions are satisfied, a transformational rule ($X \rightarrow X_1 \ldots X_n$, $f(1)...f(n)$) can be inferred.

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 Figure 2 have a transformational rule). Their corresponding CFG rule applications are marked as untransformed and are not used in training the LPCFG.

The probability of a transformational rule is computed using the maximum likelihood estimate:

$$P(LHS \rightarrow RHS, RS) = \frac{Count(LHS \rightarrow RHS, RS)}{Count(LHS \rightarrow RHS)}$$

In training the LPCFG, a large 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

Table 1. Morphological features

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 2. Corpora

Corpus	Sentence pairs	Average sentence length		Tokens		Token types	
		Eng	Viet	Eng	Viet	Eng	Viet
Computer	8718	20	21.5	173442	187138	8829	7145
Conversation	16809	8.5	8	143373	130043	9314	9557

Table 4. Data sets

Corpus	Training set	Development test set	Test set
Computer	8118	251	349
Conversation	15734	403	672

Collins' Grammar Model 1 are described in (Bikel, 2004).

3.3 Applying

After it has been trained, the transformational model is used in Step 2 of the preprocessing procedure (Table 1) for a 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). Next, the best sequence of transformational rules is computed by formula (2). Finally, by applying transformational rules to the source tree, the best transformed tree is generated.

4 Morphological Transformation

In this research, we restricted morphological analysis to the inflectional phenomenon[6]. 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[7]). 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:

Source sentence: He has traveled 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 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.

5 Experiments

5.1 Corpora and Tools

For experiments, we used two corpora (Vietnamese text has been segmented) as shown in Table 3. Computer is a very specific corpus which is collected from various computer books. The Conversation corpus contains sentences from a number of English grammar and English conversation

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

[7] In the rest of the paper, the terms lemma or lemma transformation are used alternatively.

Table 5. BLEU scores

Corpus	Baseline	Morpho		Syntax	Morpho-Syntax	
		Lemma	Pseudo word		Lemma	Pseudo word
Computer	45.12	45.41	46.68	47.85	47.16	49.57
Conversation	33.85	34.76	34.17	36.45	36.79	37.13

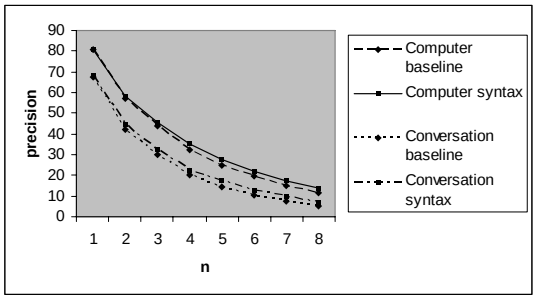

Figure 3. N-gram precisions

books. The Conversation corpus is more diverse in sentence form than the Computer corpus. These corpora are split into the training sets, the development test sets, and the test sets (Table 4).

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 system which is available for research purposes. For word alignment, we used GIZA++ tool (Och and Ney, 2000). For training a Vietnamese language model, we used SRILM tool (Stolcke, 2002). For MT evaluation, we used the BLEU measure (Papineni et al., 2001) calculated by the NIST script version 11b. For the parsing task, we used Charniak's parser (Charniak, 2000). For morphological analysis, we used a rule-based morphological analyzer which is described in (Pham et al., 2003).

5.2 Main Experimental Results

The Pharaoh phrase-based system was used as a baseline. In the training phase, the default parameter settings were used. In the testing phase, the default parameter settings were also used, except distortion weight was set to 0.5. We manually tried a relatively large number of other parameter settings using the development test sets, and found that those settings are most suitable. Experimental results on the test sets are shown in Table 5. The table shows the BLEU scores of the baseline system and other systems, which are formed by the

Table 6. Example of better translations

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 tạo_ra các ký_tự chữ_cái và chữ_số muốn ra bởi Syntax: bởi việc đẩy các kim ra theo những tổ_hợp khác_nhau , đầu in có_thể tạo_ra các ký_tự chữ_số và chữ_cái

Table 7. The statistics from learning transformational rules

Corpus	CFG rules	Groups of transformational rules	Ambiguous groups
Computer	4779	3702	951 (25.7%)
Conversation	3634	2642	669 (25.3%)

combination of the baseline 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 the differences between these corpora. Column 3 (the large one) shows 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 on the large part of the Conversation corpus which contains spoken sentences. Therefore the

Table 8. Sign tests

Corpus	Subsets	Morpho		Syntax	Morpho+Syntax		Critical value
		Lemma	Pseudo word		Lemma	Pseudo word	
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

elimination of morphological features (by lemma transformation) in the Conversation corpus is less harmful[8] than in the Computer corpus. Column 4 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 large column (Column 5) shows the scores of morpho-syntactic combinations. The combination of lemma and syntax is not good because the score for the Computer corpus is under the score of syntax alone, and for the Conversation corpus, its 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.

5.3 Some Analyses of the Performance of Syntactic Transformation

Figure 3 displays individual ngram precisions when syntactic transformation is used. Unigram precisions increase less than the others. Those 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. Table 6 contains some examples of better translations generated by the system using syntactic transformation.

Table 7 shows the statistics concerning learning transformational rules. For both corpora, the number of transformational rules which have been learned is smaller than the corresponding number of CFG rules. This is because of the sparse data problem and because there are CFG rules requiring nonlocal transformation[9]. In each corpus, the percentage of ambiguous groups is over 25%.

5.4 Significance Tests

We chose the sign test[10] (Lehmann, 1986) to test the statistical significance of our results. 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 these subsets individually. The translation systems with preprocessing were 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 is statistically significant. The same experiments were carried out for the other systems on both the Computer test set and the Conversation test set. The results are shown in Table 8 in which entries in italics are statistically significantly better than the baseline.

In column 3, 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 4 and 5).

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.

6 Conclusion

We have demonstrated that preprocessing can improve English-Vietnamese phrase-based SMT significantly, and the combination of morpho-

[8] There is a trade-off between the harmfulness (of the lost information) and the benefit (of the smoothness).

[9] That is carried out by reordering subtrees instead of CFG rules

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

syntactic transformation can achieve a better result than can be obtained with either individually. For syntactic transformation, we have proposed a transformational model based on Bayes' formula and a technique for inducing transformational rules from source-parsed bitext. Our method can be applied for other language pairs, especially when the target language is poor in resources. The use of small corpora was a limitation in our work. If larger corpora are available, more experiments should be carried out. In the future, we would like to apply this approach to other language pairs in which the difference in word order is greater than that in English-Vietnamese.

References

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 1999.

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

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

E. Charniak. 2000. A maximum entropy inspired parser. In *Proceedings of HLT-NAACL 2000*.

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

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing.* PhD Thesis, University of Pennsylvania.

M. Collins, P. *Koehn*, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL 2005*.

S. Goldwater and D. McClosky. 2005. Improving statistical MT through morphological analysis. In *Proceedings of EMNLP 2005*.

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings* of ACL 2003.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*.

P. Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *AMTA 2004*.

Y. Lee. 2004. Morphological analysis for statistical machine translation. In *Proceedings of NAACL 2004*.

E. L. Lehmann. 1986. *Testing Statistical Hypotheses (Second Edition)*. Springer-Verlag.

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

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Buildind a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics*, 19: 313-330.

I. D. Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of ACL 2004*.

Pham H. N., 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*.

S. Niessen and H. Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *Proceedings of COLING 2000*.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL 2000*.

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

F. J. Och, 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 2004*.

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

L. Shen, A. Sarkar, and F. J. Och. 2004. Discriminative reranking for machine translation. In Proceedings of HLT-NAACL 2004.

A. Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit", in *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002

Nguyen P. T., Nguyen V. V. and Le A. C.. 2003. Vietnamese Word Segmentation Using Hidden Markov Model. *International Workshop for Computer, Information, and Communication Technologies in Korea and Vietnam.*

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