Reordering Phrase-Based Machine Translation over Chunks

Vinh Van Nguyen, Thai Phuong Nguyen, Akira Shimazu and Minh Le Nguyen
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, 923-1292, JAPAN
{vinhnv, thai, shimazu, nguyenml}@jaist.ac.jp

***Abstract*—The paper presents a new method for reordering in phrase based statistical machine translation (PBMT). Our method is based on previous chunk-level reordering methods for PBMT. First, we parse the source language sentence to a chunk tree, according to the method developed by [16]. Second, we apply a series of transformation rules which are learnt automatically from the parallel corpus to the chunk tree over chunk level. Finally, we integrate a global reordering model directly in a decoder as a graph of phrases, and solve the overlapping phrase and chunk problem. The experimental results with English-Vietnamese pairs show that our method outperforms the baseline PBMT in both accuracy and speed.**

I. Introduction

In machine translation, the reordering problem (global reordering) is one of the major problems, since different languages have different word order requirements. The statistical machine translation task can be viewed as consisting of two subtasks: predicting the collection of words in a translation, and deciding the order of the predicted words (reordering problem). Currently, phrase-based statistical machine translation [6], [12] is the state-of-the-art of SMT and uses widely distance-based reordering constraints such as IBM constraints [20], ITG constraints [17], [20] and distortion limit [6]. Ideally, a model should allow reordering of any distance, because if we are to translate from Japanese to English, the verb in the Japanese sentence must be moved from the end of the sentence to the beginning just after the subject in the English sentence. With these models, phrase based SMT usually is powerful in word reordering within short distance, however, long distance reordering is still problematic.

In order to tackle the long distance reordering problem, in recent years, huge research efforts have been conducted using syntactic information. [1] shows significant improvement by keeping the strengths of phrases, while incorporating syntax into SMT. Some approaches have been applied at the word-level [2]. They are particularly useful for language with rich morphology, for reducing data sparseness. Other kinds of syntax reordering methods require parser trees , such as the work in [14], [2], [3]. The parsed tree is more powerful in capturing the sentence structure. However, it is expensive to create tree structure, and building a good quality parser is also a hard task. All the above approaches require much decoding time, which is expensive.

The approach we are interested in here is to balance between quality of translation and decoding time. Consequently, we use an intermediate syntax between POS tag and parse tree: *chunks* and *phrases*, as the basic unit for reordering. An advantage of *chunks* is closer *phrases* in PBMT.

In this paper, we also focus on researching the ordering problem, and aim to improve both the quality of translation and computation time for decoding. Our method is a global reordering model, and based on previous chunk-level reordering methods for PBMT. First, we parse the source language sentence to a chunk tree. Second, we apply a series of transformation rules which learn automatically from the parallel corpus to the chunk tree over chunks level. Third, we integrate a global reordering model directly in decoder, as a graph of phrase and solve the overlapping phrase and chunk problem. Finally, we find the best translation sentence in this graph.

The rest of this paper is structured as follows. Section 2 reviews related works. Section 3 briefly introduces PBMT. Section 4 introduces how to apply transform rules for chunks and how to deal with overlapping phrases and chunks. Section 5 briefly introduces the steps for generating a reordering graph of phrases. Section 6 describes and discusses the experimental results. Finally, conclusions are given in Section 7.

II. Related Work

To solve the reordering problem, [4] used a lexicalized reordering model as a feature in the log linear model of PSMT. However, their experiment showed that the lexicalized reordering model is not strong enough to correctly guide long distance movements.

[2] presented the reordering model based on clause restructuring. They used this model in the preprocessing step of PBMT system. The weakness of this approach is that rewriting the input sentence whether using syntactic rules or heuristics makes hard decisions that can not be undone by the decoder because this model just apply to the preprocessing step. Hence, reordering is better handled during the search algorithm, and as part of the optimization function.

[18], [19] applied Maximum Entropy (ME) model for phrase reordering. They used ME for estimating distortion probability. However, estimation is local, because the next phrase only depends on the current phrase. So, as a result, their systems are are not robust to unseen phrases.

Several methods proposed use syntactic information to handle the reordering problem. Methods by [14], [3], [8], include tree-to-string translation rules extracted from parallel corpus with linguistic annotations. However, there are some problems with linguistically syntax-based models. The first one is the expense of computational time for decoding, because the source sentence or target sentence must be parsed to a tree. The second problem is that tree-to-string rules fail for non-syntactic phrase pairs (phrase pairs that are not subsumed by any syntax tree fragments (subtree)) because they require a syntax tree fragment over the phrase to be parsed. For example: a phrase pair for English - Japanese: "`the teacher is`" and "`sensei wa`" is a non-syntactic phrase pair, because "`the teacher is`" and "`sensei wa`" are not subsumed by syntax subtree.

Note that these models have radically different structures and parameterizations than phrase-based models for PBMT.

[9] proposed a strategy to reorder a source sentence using rules based on syntactic chunks. This strategy demonstrated promising results when compared with the state of the art phrase-based system [6], in particular regarding computational time. Their strategy only reordered the phrases within each chunks of sentence, however. In other words, the chunks of a sentence were not reordered.

III. Brief description of the baseline Phrase-based SMT

In this section, we will describe the phrase-based SMT system which was used for the experiments.

Phrase-based SMT, as described by [6] translates a source sentence into a target sentence by decomposing the source sentence into a sequence of source phrases, which can be any contiguous sequences of words (or tokens treated as words) in the source sentence. For each source phrase, a target phrase translation is selected, and the target phrases are arranged in some order to produce the target sentence. A set of possible translation candidates created in this way is scored according to a weighted linear combination of feature values, and the highest scoring translation candidate is selected as the translation of the source sentence. Symbolically,

$$\hat{t} = \operatorname*{argmax}_{t,a} \sum_{i=1}^{n} \lambda_i f_j(s,t,a) \tag{1}$$

where s is the input sentence, t is a possible output sentence, and a is a phrasal alignment that specifies how t is constructed from s, and $\hat{t}$ is the selected output sentence. The weights λ_i associated with each feature f_i are tuned to maximize the quality of the translation hypothesis selected by the decoding procedure that computes the argmax.

The log-linear model is a natural framework to integrate many features. The baseline system uses the following features:

- the probability of each source phrase in the hypothesis given the corresponding target phrase.
- the probability of each target phrase in the hypothesis given the corresponding source phrase.
- the lexical score for each target phrase given the corresponding source phrase.
- the lexical score for each source phrase given the corresponding target phrase.
- the target language model probability for the sequence of target phrase in the hypothesis.
- the word and phrase penalty score, which allow to ensure that the translation do not get too long or too short.
- the distortion model allows for reordering of the source sentence.

The probabilities of source phrase given target phrases, and target phrases given source phrases, are estimated from the bilingual corpus.

[6] used the following distortion model (reordering model), which simply penalizes non-monotonic phrase alignment based on the word distance of successively translated source phrases with an appropriate value for the parameter α:

$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|} \tag{2}$$

IV. Reordering Over Chunks

A. The approach

We will extend the strategy of [9] to our new model. We will solve reordering over chunks in PBMT as the global reordering model. First, we parse the source language sentence to a chunk tree. Second, we apply the series of transformation rules which are learnt automatically for the parallel corpus to the chunk tree over chunk level. Finally, we integrate a global reordering model directly in the decoder as a graph of phrases, and find the best translation sentence in this graph. When we integrate a global reordering model in the decoder to create a phrase graph, we must solve the overlapping phrase and chunk problem.

Our approach is similar to [21] except for the following important differences: first, we parse the source language sentence to a chunk tree, while they parse the source use chunking. Second, we use transformation rules with a hierarchial structure, so we will reorder over chunks more generally, while they use the rules without hierarchical structure. Finally, we solve the overlapping phrase and chunk problem, while they do not mention this problem.

B. The algorithm for overlapping phrase and chunk problem

In this section, we will describe a heuristic algorithm for solving a problem of overlapping phrase and chunk and generating a graph of phrases. We conduct error analysis of the translation output of the "Over Chunks" system and observe that phrases which overlap chunks (those chunks are reordered) usually omitted in a decoding process. With an example in Section 4.2, phrase "what characteristics does" can be omitted because this phrase overload two chunks: [what characteristics] and [does] (ordering of those chunks in a target sentence is [does][what characteristics]). So, we find the solution to exploit both phrases and chunks in decoding process. With a simple idea: phrase is so close to chunk, we

Algorithm 1

Input: set of chunks ($\Delta = \{c_{kl}\}$)
set of phrases ($\Gamma = \{p_{ij}\}$)
1: Reorder(Δ)
2: **for** ($i = 0 \rightarrow n-1$)
3: **for** ($p_{ij} \in \Gamma$)
4: **for** ($c_{kl} \in p_{ij}$)
5: **if** ($k' \notin [i,j]$ or $l' \notin [i,j]$) **then**
6: $\Theta = \Theta \cup p_{ij}$
7: **for**($p_{xy} \in \Theta$)
8: Reorder($p_{xy}, c_{kl} \notin p_{xy}$)
9: **for** ($p_{xy} \in \Theta$)
10: **for** ($i = y+1 \rightarrow n-1$)
11: **for** ($p_{ij} \in \Gamma$)
12: **for** ($c_{kl} \in p_{ij}$)
13: **if** ($k' \notin [i,j]$ or $l' \notin [i,j]$) **then**
14: $\Omega = \Omega \cup p_{ij}$
15: **if** ($p_{x_1 y_1} \in \Omega$) **then**
16: Reorder($p_{xy}, p_{x_1 y_1}, c_{kl} \notin p_{xy}$ and $c_{kl} \notin p_{x_1 y_1}$)

Fig. 1. Algorithm for solving the overlapping chunks and phrases and generating a graph of phrases

reorder the phrases based on chunks approximately (chunk is "head" of phrase).

The algorithm of solving a problem of overlapping phrase and chunk first implements the reordering over all chunks, and then reorders k phrases separately based on the reordering of chunks (the algorithm is described with $k = 2$ because the algorithm takes an expensive time with $k > 2$), and generates all possible paths in a graph of phrases. The efficiency of this algorithm is showed in the Section 6.3. The algorithm is presented in Figure 1 as a Algorithm 1.

Input: A set of chunks (Δ), and a set of phrases for a input sentence (Γ).

We assume that the input sentence is represented as $w_0 \ldots w_n$ where w_i is the *i-th* word in the input sentence. We denote p_{ij} is phrase with a start position i and an end position j in a input sentence; $length(p_{ij})$ is size of phrase p_{ij} (a number of words in phrase p_{ij}); c_{kl} is the chunk with a start position k and an end position l; $c'_{k'l'}$ is a reordered chunk of a chunk c_{kl} in a reordered sentence.

Line 1 in algorithm 1, we implement a reordering over all chunks according to transformation rules to generate a possible reordered sentence. From line 2 to line 6, from left to right, we find all phrases p_{ij} ($0 \leq i < j \leq n$) in an input sentence which satisfy: at least a chunk c_{lk} which a chunk $c'_{l'k'}$ does not belong to $[l,k]$ in a reordered sentence. We consider a reordered position of c_{lk} as the reordered position of the phrase p_{ij} in a reordered sentence. We store those found phrases in a set Θ. Line 7 and line 8, we implement a reordering each phrase p_{xy} of a set Θ and remaining chunks (the chunks do not belong to p_{xy}) to generate a possible reordered sentence.

From line 9 to line 14 in algorithm 1, with each phrase p_{xy} belong to a set Θ, from left to right, we find all phrases p_{ij} ($y < i < j \leq n$ in an input sentence which satisfies: at least a chunk c_{lk} which a chunk $c'_{l'k'}$ does not belong to $[i,j]$ in a reordered sentence. We consider a reordered position of c_{lk} as the reordered position of the phrase p_{ij} in a reordered sentence. Line 15 and line 16, we implement a reordering each phrase p_{xy} of a set Θ and each phrase $p_{x_1 y_1}$ and remaining chunks (the chunks do not belong to p_{xy}) to generate a possible reordered sentence.

For example:
Input sentence: *what characteristics does the smart student have ?*
Chunks and tags: [what characteristics WHNP][does AUX][the smart student NP][have VP] [? .]
Positions of chunks: 0 1 2 3 4
Syntax tree: (SBARQ (WHNP (WP what NN characteristics)) (SQ (AUX does) (NP (DT the JJ smart NN student)) (VP (VB have))) (. ?))
(1) Position of the reordering over chunks: 23104 (using two transformation rules of English-Vietnamese: (SBARQ $\rightarrow$ WHNP SQ ?, 1 0 2) and (SQ $\rightarrow$ AUX NP VP, 1 2 0))

If we do not consider the phrases of an input sentence that overlap the chunks, we implement the reordering over chunks from an input sentence to a reordered sentence as Figure 2. So, two phrases can be omitted in decoding process: "what characteristics does" and "does the".

[the smart student NP] [have VP] [does AUX][what characteristics WHNP] [? .] (according to (1))

Words and Phrases: "what", "characteristics", "does", "the", "smart", "student", "have", "?", "what characteristics does", "does the", "smart student", "student have".

Therefore, we need solve the overlapping phrase and chunk problem. The algorithm for overlapping phrases and chunks is demonstrated in Figure 2. We denote a black line as a chunk and a dotted black line such as a phrase. We begin with phrase "what characteristics does" because this phrase includes two chunks: [what characteristics WHNP][does AUX], where chunk [what characteristics WHNP] satisfies a reordered position do not belong to an interval [0, 2] in the reordered sentence. Consequently, we consider the reordering of chunk [what characteristics WP] as the reordering of the phrase "what characteristics does". We implement a reordering from a phrase "what characteristics does" and chunks [the smart student NP], [have VP], and [?]. We have a possible reordered sentence showed in Figure 2: *[the smart student] [have] "what characteristics does" [?]*. We implement similarly a reordering of a phrase "does the" and "student have".

The Figure 4 shows a part of a graph of phrases after reordering with the above example.

V. Reordering Graph Generation

A. Parsing the Source Sentence

First, a POS tagger is usually used for chunk parsing. In our experiments, we used the tagger tool which is based on CRFs [7] then we used chunkparser-1.0 [16] to parse an English sentence to a tree. The main advantage of this method not only is fast computation time but also the accuracy of that was about 85% with F1 score.

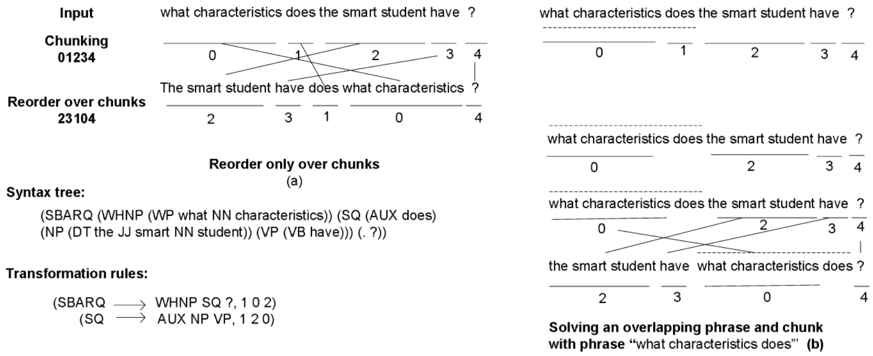

Fig. 2. Example for solving an overlapping Phrases and Chunks

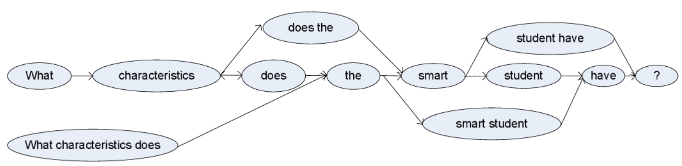

Fig. 3. A graph of phrases before reordering

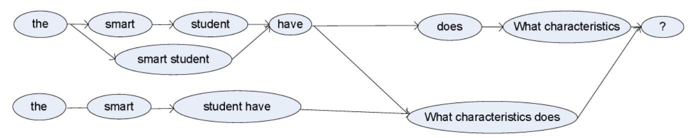

Fig. 4. A part of a graph of phrases after reordering

B. Transformation Rules

Suppose that T_s is a given lexicalized tree of the source language (whose nodes are augmented to include a word and a POS label). T_s contains n applications of lexicalized CFG rules $LHS_i \rightarrow RHS_i$ ($i \in \overline{1,n}$). We want to transform T_s into the target language word order by applying transformational rules to the CFG rules. A transformational rule is represented as ($LHS \rightarrow RHS, RS$), which is a pair consisting of an unlexicalized CFG rule and a reordering sequence (RS). For example, the rule (NP $\rightarrow$ JJ NN, 1 0) implies that the CFG rule (NP$\rightarrow$ JJ NN) in the source language can be transformed into the rule (NP$\rightarrow$NN JJ) in the 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 (we can see [10] for a description in more detail).

We use the method described in [10] to extract the transformation rules from the parallel corpus and induce the best sequence of transformational rules for a source tree.

C. Applying transformation rules

First, we apply a series of transformation rules to the source tree for reordering over chunks. Next, we use the method described in Section 4.2 for solving the overlapping of phrases and chunks. Finally, we generate a reordered graph of phrases, and find the best translation sentence in this graph.

D. Graph generation

For example, given a source sentence "what characteristics does the smart student have ?" in the above example in Section 4.2, we have a possible graph of phrases before reordering as showed in Fig 3.

After we apply a series of transformation rules (two rules: (SBARQ $\rightarrow$ WHNP SQ ?, 1 0 2); (SQ $\rightarrow$ AUX NP VP, 1 2 0)) and solve the overlapping of phrases and chunks to the above example, we have a part of a possible phrase graph after reordering as shown in Figure 4.

All reorderings of an input sentence are encoded and stored in a phrase graph. Each path is a possible reordering S' and given a probability P. In this paper, the probability is computed using the transformation probability of the syntactic transformation model [10].

VI. Experiment

A. Implementation

- We used chunkparser-1.0 [16] to parse a source sentence (English sentence) to a tree.
- The rules are learnt from English-Vietnamese parallel corpus and Penntree Bank Corpus. We used the CFG transformation rules (chunk level) for being extracted from [10]'s method for reordering over chunks of an input sentence.
- Design of decoding is adapted from Moses [5]. In decoding, integration of an input sentence is handled as a graph of phrases.

B. Data sets

We conducted the experiments with English-Vietnamese pairs. We used two English-Vietnamese corpora, one was collected from some grammar books (named "Conversation") and other one collected from daily newspapers (named "General"). These corpora, which include 16809 sentences and 55341 sentences for "Conversation" and "General", respectively, are split into training sets, development test sets, the test sets. The

TABLE I
CORPORA AND DATA SETS (SENTENCES)

Corpus	Sentence pairs	Training set	Dev set	Test set
Conversation	16809	15734	403	672
General	55341	54642	200	499

TABLE II
STATISTICAL RESULTS OF REORDERING SENTENCES IN ENGLISH SENTENCES

Corpus	Sentences	Sentences with reordering
Conversation	672	215 (31.99%)
General	499	149 (29.86%)

TABLE IV
TRANSLATION TIME FOR THE ENGLISH-VIETNAMESE "CONVERSATION" TEST SET

Method	Computation time	Sec per sen
Baseline	1489 sec	2.2 sec
Over Chunks	597 sec	0.88 sec

statistical information in detail about three corpora is shown in Table 1.

We tested 672 English sentences (test set of Conversation Corpus English-Vietnamese) and 499 English sentence (test set of General Corpus English-Vietnamese) for using CFG transformation rules (level over chunk). The result of statistics is showed on Table 2. A number of sentences which really are reordered over chunk level are 215, by 31.99 % and 149, by 29.86 % with "Conversation" and "General", respectively. This result also shows that the problem of the reordering over chunk level is very important with the language pairs that have the difference grammar structures, such as English-Vietnamese language.

C. BLEU score and computational time

We carried out the experiments on a PC with Pentium IV processor 2GHz, RAM memory 1GB. We ran GIZA++ [11] on the training corpus in both directions using its default setting, and applied the refinement rule "diag-and" [6] to obtain a single many-to-many word alignment for each sentence pair. For learning language models, we used the SRILM toolkit [15]. For MT evaluation, we used the BLEU measure [13] calculated by the NIST script version 11b.

The translation results are presented in Table 3. The baseline system is a non-monotone translation system, in which the decoder does reordering on the target language side (we adapted the beam search decoding algorithm [5]). The BLEU score of "Over Chunks" and "Over Chunks + Overlapping" systems (OOC) are 36.12% and 36.73% absolute, which improved by 0.46% and 1.07% compared with the baseline of "Conversation". The BLEU score of "Over Chunks" and "Over Chunks + Overlapping" systems are 34.69% and 35.22% absolute, which improved by 0.62% and 1.15% compared with the baseline of "General". Table 3 also shows the effect of a overlapping phrases and chunks. The "Over Chunks + Overlapping" systems improved by 0.61% and 0.53% compared with "Over Chunks" only systems of "Conversation" and "General", respectively. An improvement of "Overlapping" is well worthwhile. Those values showed that: (1) the improvement is higher with language pairs which are more different in word order; (2) PBMT captures reordering quite well if there is a large amount of training.

After we implemented the reordering phrase over chunks, we used the method described in [9] to reorder in each chunk of our system, named "Over Chunks+Overlapping+In Chunks". The results are also shown in Table 3 which outperform that of OOC by 0.96% and 1.08 % absolute with "General" and "Conversation", respectively.

Though the input is a graph, the source reordering is still faster than the reordering during decoding. We conducted the experiment with "Conversation" corpus. The results are showed in Table 4. The baseline system took 2.2 seconds per sentence and the "Over chunks" took 0.88 seconds per sentence. In short, the decoding time of our method is faster than that of baseline, by the approximate factor of 3 with the "Conversation" corpus.

VII. CONCLUSION

In this paper, we have presented a new method for reordering in phrase based statistical machine translation (PBMT). The experimental results with English-Vietnamese pairs show that our method outperforms the baseline PBMT in both accuracy and speed.

In future, we will solve the overlapping phrase and chunk problem generally, and more effectively.

ACKNOWLEDGMENT

We would like to thank to anonymous reviewers for helpful discussions and comments on the manuscript. The work on this paper was supported by the JAIST 21 century COE program "Verifiable and Evolvable e-Society".

REFERENCES

[1] D. Chiang,"A hierarchical phrasebased model for statistical machine translation". *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL05)*, pages 263-270. Association for Computational Linguistics, Ann Arbor, Michigan. 2005

[2] M. Collins, P. Koehn, and I. Kucerova, "Clause restructuring for statistical machine translation". *In Proc. ACL 2005*, pages 531-540. Ann Arbor, USA. 2005

TABLE III
TRANSLATION PERFORMANCE FOR THE ENGLISH-VIETNAMESE TASKS

Corpus	Method	BLEU score[%]
Conversation	Baseline	35.66
	Over Chunks	36.12
	Over Chunks + Overlapping (OOC)	36.73
	Over Chunks + Overlapping + In Chunks	37.81
General	Baseline	34.07
	Over Chunks	34.69
	Over Chunks + Overlapping (OOC)	35.22
	Over Chunks + Overlapping + In Chunks	36.18

[3] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer, "Scalable inference and training ofcontext-rich syntactic translation models". *In Proceedings of COLING/ACL 2006*, pages 961-968. Sydney, Australia.

[4] Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. 2005. "Edinburgh system description for the 2005 NIST MT evaluation". *In Proceedings of Machine Translation Evaluation Workshop 2005*.

[5] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constrantin, and Evan Herbst, "Moses: Open source toolkit for statistical machine translation". *In Proceedings of ACL, Demonstration Session*. 2007

[6] Philipp Koehn, Franz Josef Och, and Daniel Marcu, "Statistical phrase-based translation". *In Proceedings of HLT-NAACL 2003*, pages 127-133. Edmonton, Canada. 2003

[7] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". *In Proc. 18th International Conference on Machine Learning*, pages 282-289. Morgan Kaufmann, San Francisco, USA. 2001

[8] Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight, "Statistical machine translation with syntactified target language phrases". *In Proceedings of EMNLP 2006*, pages 44-52. Sydney,Australia. 2006.

[9] Phuong Thai Nguyen, Akira Shimazu, Le-Minh Nguyen, and Van-Vinh Nguyen, "A syntactic transformation model for statistical machine translation". *International Journal of Computer Processing of Oriental Languages (IJCPOL)*, 20(2):1-20. 2007.

[10] Thai Phuong Nguyen and Akira Shimazu, "Improving phrase-based smt with morphosyntactic analysis and transformation". *In Proceedings AMTA*. 2006.

[11] Franz Josef Och, Hermann Ney, "A Systematic Comparison of Various Statistical Alignment Models". *Computational Linguistics*, volume 29, number 1, pp. 19-51 March 2003

[12] Franz J. Och and Hermann Ney, "The alignment template approach to statistical machine translation". *Computational Linguistics*, 30(4):417-449. 2004.

[13] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002, "Bleu: a method for automatic evaluation of machine translation". *In Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311-318, Philadelphia, PA, July.

[14] Chris Quirk, Arul Menezes, and Colin Cherry. 2005. "Dependency treelet translation: Syntactically informedphrasal SMT". *In Proceedings of ACL 2005*, pages 271-279. Ann Arbor, Michigan, USA.

[15] Andreas Stolcke, "Srilm - an extensible language modeling toolkit". *In Proceedings of International Conference on Spoken Language Processing*, volume 30, pages 901-904. 2002.

[16] Yoshimasa Tsuruoka and Junchi Tsujii. 2005. "Chunk parsing revisited". *In Proceedings of the 9th International Workshop on Parsing Technologies (IWPT 2005)*.

[17] D. Wu. 1996. "A polynomial-time algorithm for statistical machine translation". *In In Proceedings of ACL96*, pages 152-158. Santa, Cruz, CA.

[18] Deyi Xiong, Qun Lui, and Shouxun Lin. 2006. "Maximum entropy based phrase reordering model for statistical machine translation". *In Proceedings of ACL06*, pages 521-528.

[19] Richard Zen and Hermann Hey. 2006. "Discriminative reordering models for statistical machine translation". *In Proceeding of the Workshop on Statistical Machine Translation*, pages 55-63.

[20] R. Zens, H. Ney, T. Watanabe, and E. Sumita, "Reordering constraints for phrase-based statistical machine translation". *In Proceedings of the 20th International Conference on Computational Linguistics (CoLing)*, pages 205-211. Geneva, Switzerland. 2004.

[21] Yuqi Zhang, Richard Zens, and Hermann Ney, "Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation". *In Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1-8. 2007