Simple but Effective Approaches to Improving Tree-to-tree Model

Feifei Zhai, Jiajun Zhang, Yu Zhou and Chengqing Zong
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{ffzhai, jjzhang, yzhou, cqzong}@nlpr.ia.ac.cn

Abstract

Tree-to-tree translation model is widely studied in statistical machine translation (SMT) and is believed to be much potential to achieve promising translation quality. However, the existing models still suffer from the unsatisfactory performance due to the limitations both in rule extraction and decoding procedure. According to our analysis and experiments, we have found that tree-to-tree model is severely hampered by several rigid syntactic constraints: the both-side subtree constraint in rule extraction, the node constraint and the exact matching constraint in decoding. In this paper we propose two simple but effective approaches to overcome the constraints: utilizing fuzzy matching and category translating to integrate bilingual phrases and using head-out binarization to binarize the bilingual parsing trees. Our experiments show that the proposed approaches can significantly improve the performance of tree-to-tree system and outperform the state-of-the-art phrase-based system Moses.

1 Introduction

In recent years, syntax-based translation models have shown promising progress in improving translation quality. These models include ***string-to-tree models*** (Galley et al., 2006; Marcu et al., 2006; Shen et al., 2008; Chiang et al., 2009), ***tree-to-string models*** (Quirk et al., 2005; Liu et al., 2006; Huang et al., 2006; Mi et al.,2008), and ***tree-to-tree models*** (Eisner, 2003; Ding and Palmer, 2005; Cowan et al., 2006; Zhang et al., 2008; Liu et al., 2009). With the ability to incorporate both source and target syntactic information, tree-to-tree models are believed to be much potential to achieve promising translation quality. However, the conventional tree-to-tree based translation systems haven't shown superiority in empirical evaluations.

To explore the reasons why tree-to-tree model is so unsatisfactory, this paper makes a deep analysis of the limitations on its rule extraction and decoding procedure respectively.

Towards rule extraction, we found that in our training corpus the bilingual phrases that tree-to-tree model can cover only account for 8.45% of all phrases due to the ***both-side subtree constraint***. This low proportion definitely causes a severe poor rule coverage problem and leads to a bad translation quality.

What's more, in decoding phase, the decoding space is severely limited by the ***node constraint*** and the ***exact matching constraint***, which makes the search space too narrow to get a promising result.

Obviously, tree-to-tree model is profoundly affected by the rigid syntactic constraints. In order to resolve the constraints, two simple but very effective approaches are proposed straightforwardly in this paper: 1) integrating bilingual phrases to improve the rule coverage problem; 2) binarizing the bilingual parsing trees to relieve the rigid syntactic constraints.

The paper is structured as follows. Section 2 carefully analyzes the limitations of tree-to-tree model and introduces the currently existing improvements on tree-to-tree model. Section 3 elaborates the proposed approaches in more details. In Section 4, we evaluate the effectiveness of our approaches and finally conclude with a summary and our future work in Section 5.

2 Analysis on Tree-to-tree Model

Given a word-aligned tree pair $T(f_1^I)$ and $T(e_1^J)$ as shown in Fig.1, a tree-to-tree rule r is a pair of aligned subtrees derived from the tree pair:

$$r = < ST\,(\,f_{i_1}^{i_2}\,),\, ST\,(e_{j_1}^{j_2}\,),\, \tilde{A} > \qquad (1)$$

Where, $ST(f_{i_1}^{i_2})$ is a source subtree, covering span $[i_1, i_2]$ in $T(f_1^I)$; $ST(e_{j_1}^{j_2})$ is a target subtree, covering span $[j_1, j_2]$ in $T(e_1^J)$; $\tilde{A}$ is the alignment between leaf nodes of the two subtrees, satisfying the constraints: $\forall(i,j)\in \tilde{A}: i_1 \le i \le i_2 \leftrightarrow j_1 \le j \le j_2$. The leaf nodes of a subtree can be either non-terminal symbols (grammar categories) or terminal symbols (lexical words). Fig.2 shows two example rules extracted from the tree pair in Fig.1.

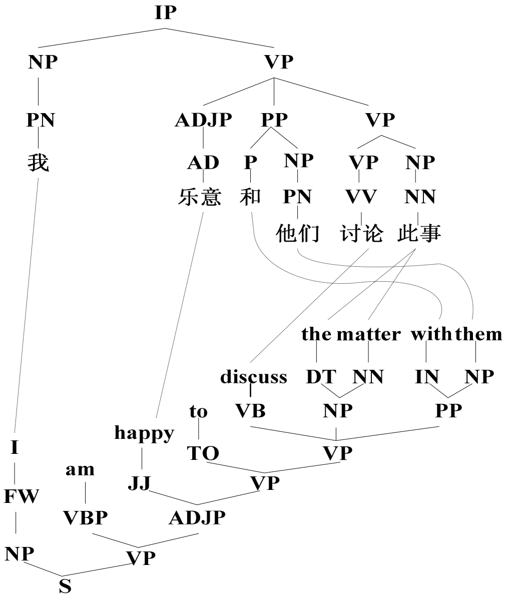

Figure 1. An example of Chinese-English tree pair.

2.1 Limitations on Tree-to-tree Rule Extraction

To extract all valid tree-to-tree rules, (Liu et al., 2009) extends the famous tree-to-string rule extraction algorithm GHKM (Galley et al., 2004) to their forest-based tree-to-tree model. However, only with GHKM rules, the rule coverage is very low[1]. As SPMT rules (Marcu et al., 2006) have proven to be a good complement to GHKM (DeNeefe et al., 2007), we also extract full lexicalized SPMT rules to improve the rule coverage.

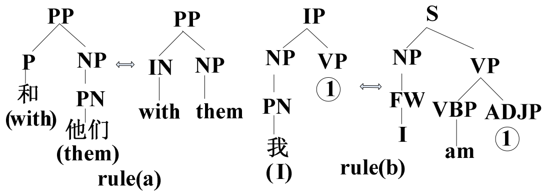

Figure 2: Two examples of tree-to-tree rule.

The tree-to-tree style SPMT algorithm used in our experiments can be described as follows: for each phrase pair, traverse the source and target parsing tree bottom up until it finds a node that subsumes the corresponding phrase respectively, then we can extract a rule whose roots are the nodes just found and the leaf nodes are the phrases.

However, even with GHKM and SPMT rules, the rule coverage is still very low since tree-to-tree model requires that both source side and target side of its rule must be a subtree of the parsing tree. With this hard constraint (Liu et al., 2009; Chiang, 2010), the model would lose a large amount of bilingual phrases which are very useful to the translation process (DeNeefe et al., 2007).

Eng Chn	tree	non-tree	total
tree	1.24M (8.45%) *(t2t, s2t, t2s,pb)*	3.19M (21.75%) *(t2s, pb)*	4.43M (30.2%)
non-tree	1.5M (10.24%) *(s2t, pb)*	8.74M (59.56%) *(pb)*	10.24M (69.8%)
total	2.74M(18.69%)	11.93M(81.31%)	14.67M

Table 1: Distribution of the bilingual phrases in Chinese-to-English FBIS corpus (LDC2003E14). In each cell, the first line represents the number of the bilingual phrases and the percentage it accounts for. The second line lists all models that can get the corresponding bilingual phrases as rules[2].

Table 1 shows the distribution of the bilingual phrases in Chinese-English FBIS corpus. The max length of the bilingual phrases is constrained to 7. In the table, rows and columns denote whether the source (Chinese) phrase and target (English) phrase correspond to subtree respectively. From the table, we can easily conclude that phrase-based model can extract all useful phrase pairs, while string-to-tree and tree-to-string model can only extract part of them because of the one-side subtree constraint. Further, with the rigid ***both-side subtree constraint***, rule space of tree-to-tree model is the narrowest one which can only account for at most

[1] (Liu et al., 2009) investigate how many phrase pairs can be captured by full lexicalized tree-to-tree rules. They set the maximal length of phrase pairs to 10 and the maximal node count of tree-to-tree rule was set to 10. For the tree-to-tree model, the coverage was below 8%. Even with packed forest, the coverage was only 9.7%.

[2] The abbreviations correspond to different translation models: ***t2t***: tree-to-tree model; ***t2s***: tree-to-string model; ***s2t***: string-to-tree model; ***pb***: phrase-base model.

8.45% of all phrase pairs. Hence, how to learn from other models to enlarge the rule coverage is a big problem for tree-to-tree model.

2.2 Limitations on Tree-to-tree Decoding

In the decoding procedure, tree-to-tree model traverses the source parsing tree bottom up and tries to translate the subtree rooted at the current node. If the employed rule is full lexicalized, *candidate translations*[3] are generated directly, otherwise new candidate translations are created by combining target terminals of the rule and candidate translations of the corresponding descendant nodes of the current node. Root node of the parsing tree will be the last visited node and the final best translation can be got from its candidate translations.

Broadly, tree-to-tree based decoding is node-based, i.e., only the source spans governed by tree nodes can be translated as a unit. We call these spans ***translation spans***. For example, in Fig.1, span "*和 他们*" is a translation span because it is governed by node PP, while span "*和 他们 讨论 此事*" is not a translation span since none subtree corresponds to it. During decoding, translation spans are used for translation, while other spans are ignored completely even if they include better translations. Thus this rigid constraint (we call it ***node constriant***) will exclude many good translations.

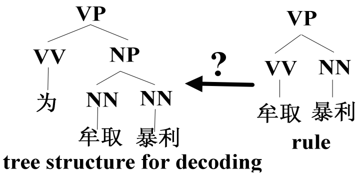

Figure 3. Mismatch because of parsing errors.

If we take the Chinese part of the abovementioned FBIS corpus as a test set, the source (Chinese) phrase that has a counterpart phrase on the target (English) side in terms of the word alignment can serve as an effective translation span of the current sentence because it has effective translations. In our statistics, there are in total of 14.68M effective translation spans in the corpus. However, only 44.6% (6.54M spans) of them are governed by tree nodes. This low proportion would definitely lead to a really narrow search space for tree-to-tree model and further a poor translation quality.

In addition, the model is also heavily affected by the ***exact matching constraint*** which means only the rules completely matching part of the source tree structure can be used for decoding. Fig.3 shows an example of this constraint. In Fig.3, there is a parsing error on the tree structure of phrase "*牟取 暴利*" which leads to a mismatch between the rule and tree structure of the test sentence. Since the parsing error is very common with automatic parsers, the mismatch cannot be a rare phenomenon. Moreover, the large and flat structures which have a close relation with reordering are also hard to match exactly. Thus with such constraint, many rules cannot be employed during decoding even if they are extracted by the model and the search space would necessarily be decreased. Hence, how to find a good way to extend the search space is another big problem for tree-to-tree model.

2.3 Related Work to Improve Tree-to-tree Model

Two main directions have been emerged to overcome the limitations discussed above.

One is to loose the syntactic constraints. (Zhang et al., 2008) proposes a *tree-sequence based tree-to-tree model* that represents rules with tree sequences and takes all spans as translation spans. This method resolves the both-side subtree constraint and the node constraint thoroughly, but it neglects the bad influence of the exact matching constraint. Furthermore, it is obviously that each bilingual phrase would multiply into many tree sequence rules with different structures, which definitely leads to serious rule expansion to increase the decoding burden.

In the other direction, more information is introduced into the model. (Liu et al., 2009) substitutes 1-best tree with packed forest for tree-to-tree model which can compactly encode many parses and successfully relieve the constraints, but even with packed forest, the rule coverage is still very low[4].

The two directions have proven to outperform their conventional counterparts significantly. However, no matter tree sequence or packed forest, they are all complicated to deal with in decoding stage,

[3] A candidate translation is a target subtree with some real property values for decoding, e.g. language mode.

[4] Please refer to footnote 1.

and furthermore, they both need to modify the conventional tree-to-tree model, thus the original decoding algorithm must be immensely adjusted to cater for the corresponding changes.

3 Our Approaches

Different with the existing related work, aiming at resolving the rigid syntactic constraints more directly and essentially, we propose simple but very effective approaches to improve the conventional tree-to-tree model: integrating bilingual phrases and binarizing the bilingual parsing trees.

3.1 Integrating Bilingual Phrases

Inspired by (Liu et al., 2006) and (Mi et al., 2008) on utilizing bilingual phrases to improve tree-to-string and forest-to-string model, we integrate bilingual phrases into tree-to-tree model to resolve the problem of poor coverage of rules, which is more difficult since we have to provide syntactic structures for both the source and target phrases to serve the decoding process of the model. Here we present two simple approaches to transform the source and target phrases into tree-to-tree style rules respectively. After that, all bilingual phrases are integrated into the model easily.

3.1.1 Source Phrase Transformation

In traditional tree-to-tree based decoding, source side of the rule is employed to match the source parsing tree exactly. Thus if we want to use a source phrase, theoretically, we must decorate it with the corresponding syntax structure like tree-sequence based model. However, our analysis has shown that exact match would do harm to the translation quality. Thus instead of syntax structures, we decorate the source phrases with proper syntactic categories which have been proven to be necessary and effective for translation (Zhang et al., 2011b). When decoding with these source phrases, we ignore the internal structure of the subtree for translation and only match the rule's category with root node of the subtree along with the matching between leaf nodes, just as shown in Fig.4.

Here we utilize the SAMT grammar (Zollmann and Venugopal, 2006), with which each phrase can be associated with a corresponding syntactic category. For example, in Fig.1 the source span "*和 他们 讨论 此事*" does not correspond to a subtree, but we can annotate an SAMT category PP*VP[5] for it and generate rule(c). The annotation method is taken from (Zhang et al., 2011b). The details are ignored due to the space limitation of the paper.

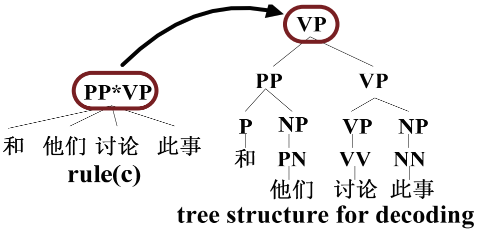

Figure 4. Rule(c) is an example of source phrase after transformation. When translating the tree structure, match the rule's category with the head node and match the rule's words with the terminal nodes of the structure. In the figure, if we do exact match between categories, rule(c) cannot be used yet.

Normally, if we do exact match, rule(c) in Fig.4 will not be employed due to the mismatch between categories of rule and tree structure. Hence, to maximize the capacities of the source phrases, we utilize fuzzy matching method which has been successfully employed in hierarchical phrase-based model (Huang et al., 2010) and string-to-tree model (Zhang et al., 2011b) to match categories.

With fuzzy matching method, we represent each SAMT-style syntactic category with a real-valued vector $\vec{F}(c)$ using latent syntactic distribution. Due to the space limitation, here we ignore the details as we just follow the work of (Huang et al., 2010; Zhang et al., 2011b). Then the degree of syntactic similarity between two categories can be simply computed by dot-product:

$$\vec{F}(c)\cdot\vec{F}(c')=\sum_{1\le i\le n} f_i(c)f_i(c') \qquad (2)$$

which yields a similarity score ranging from 0 (totally syntactic different) to 1 (totally syntactic identical).

That is to say, we transform an original source phrase by decorating it with a SAMT-style syntactic category and a corresponding real-valued vector. During decoding, we consider all possible source phrases and compute the similarity scores between categories of phrases and head nodes of the current translated structure. Then the similarity score will serve as a good feature (we call it *similarity score feature*) incorporated into the model and let it learn how to respect the source phrases.

[5] Where * is just a conjunction mark.

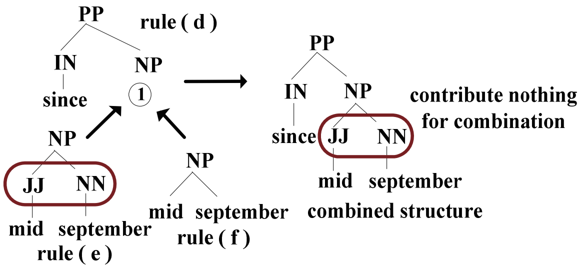

Figure 5. To combine target rules to create translation. In the figure, source side rules are discarded. The combination process is only related to the head node and leaf nodes of the rule, but has nothing to do with the internal structure. Thus, during decoding, rule(e) and rule(f) are equal.

3.1.2 Target Phrase Transformation

In decoding phase, target sides of rules are combined successively to create a syntax tree of the translation. Fig.5 shows the process of combination: the non-terminal leaf node of rule(d) (i.e. NP) is identical to the head node of rule(e), so we can combine the two rules to form a larger structure. In the whole process, the internal structures contribute nothing to combination. Thus just like source phrase, we can transform the target phrases into tree-to-tree style rules naturally only by assigning them proper syntactic categories for combination.

It is natural to assign the head node category for a target phrase if it corresponds to a subtree, but which category is proper for the phrase that does not correspond to a subtree? The easiest way is to assign all categories to the phrase and let the decoder choose the best one. But without enough information, the decoder will be confused with the choices.

As categories of the phrases are only used for further combination during decoding, we can certainly postpone the category assigning work to the decoder at the time when we obtain the information of current source node. Here, we take ***category translation probability*** $p(TC\,|\,SC)$ as our basis to choose categories which means the probability to translate source category *SC* into target category *TC*:

$$p(TC\,|\,SC) = \frac{count(SC-TC)}{count(SC)} \qquad (3)$$

In the formula, count(*SC*) are counts of *SC* in corpus and count(*SC-TC*) denotes the counts when *SC-TC* is a translation pair which requires the two nodes must be head nodes of one tree-to-tree rule (GHKM or SPMT) at least. For example, in Fig.1, PP-PP and IP-S are translation pairs.

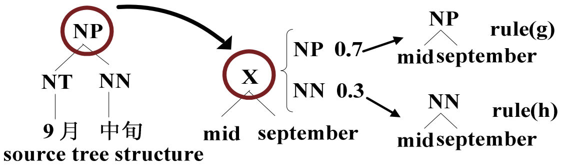

Fig.6: Transform target phrases for decoding. We assign categories by translating the category of the source node into one or many proper target categories during decoding and form one or more rules. In the figure, the category translation probability P(NP|NP) = 0.7 and P(NN|NP) = 0.3. So we can transform the original rule into rule(g) and rule(h) and the probability will serve as a feature incorporated into the model. Noting that rule(g) and rule(h) are only temporary rules, and if the original rules are employed when dealing with a different structure, many different rules might be generated.

As illustrated in Fig.6, in rule extraction process, we label target phrases[6] with a general category, Such as X, as a placeholder. When decoding with target phrases, assign categories to target phrases on-line for further combination according to the source node and *category translation probability*. We integrate all possible categories to avoid rigid choice and introduce the category translation probability into the model as a feature (we call it *category translation feature*) to punish categories with low probabilities.

3.2 Tree Binarization

Tree binarization methods have been successfully adopted to improve string-to-tree models (Wang et al.,2007, 2010) and tree-to-string models (Zhang et al., 2011a). We believe it can also deliver a promising improvement for tree-to-tree model since the syntactic constraints on tree-to-tree model are more rigid than string-to-tree and tree-to-string model.

Tree binarization means to convert the parsing trees into binary trees by introducing new tree nodes. The approach we take here is the head-out binarization (Wang et al., 2007). The children to the left of the head word are binarized in one direction, and the children to the right are binarized in the other direction. We label the newly created node with its original father node category plus a flag (e.g. VP-COMP) as illustrated in Fig.7.

[6] If the target phrase corresponds to a subtree, we will not do transformation, but annotate it with the head node category because we believe it is relatively more accurate and effective.

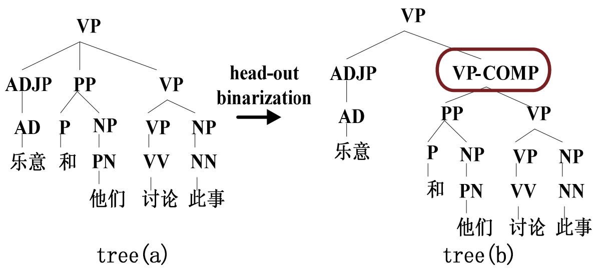

Fig.7: An example of binarization from the aligned tree pair in Fig.1.

There are three aspects for binarization to display its advantages in tree-to-tree model.

First, by introducing new nodes, it helps to overcome the problem of poor rule coverage. In our training corpus, the percentage of the bilingual phrases that tree-to-tree model can cover are increased from 8.45% (tree-tree cell in Table 1) to 11.39% after binarization, which results in a better rule coverage. As an example, in Fig.7, with the new node VP-COMP, the source span “*和 他们 讨论 此事*” can correspond to a subtree, and its counterpart at the target side is a subtree in Fig.1, thus we can extract a rule to translate “*和 他们 讨论 此事*” into “*discuss the matter with them*”.

Second, the node constraint on decoding is also resolved by the newly created nodes. In NIST MT04 and MT05 test data, 7.94 and 7.75 nodes are created by tree binarization for each sentence in average. Compared to 59.01 and 58.39 nodes per tree on average before binarization, the numbers of nodes as translation spans are increased by 13.46% and 13.27% respectively, which leads to a larger decoding space.

Third, it helps to alleviate the exact matching constraint by converting the flat and large structures into binary structures. For example, if we want to look for a rule for root node VP in Fig.7, compared to the tri-structure in tree(a), obviously, matching the binary structure of tree(b) is much simpler, which can provide more matched rules and a larger search space.

4 Experiments

4.1 Experimental Setup

The experiments are conducted on Chinese-to-English translation; the training data is FBIS corpus containing about 7.1 million Chinese words and 9.2 million English words. We perform bidirectional word alignment using GIZA++, and employ *grow-diag-final-and* strategy to generate the symmetric word alignment. We parsed both sides of the parallel corpus with the Berkeley parser (Petrov et al., 2006) and trained a 5-gram language model with the Xinhua portion of English Gigaword corpus.

For tuning and testing, we use NIST MT evaluation data for Chinese-to-English from 2003 to 2005 (MT03 to MT05). The development data set comes from MT03 in which sentences with more than 20 words are removed to speed up MERT (Och, 2003). The test set includes MT04 and MT05.

The baseline tree-to-tree system is implemented by ourselves according to (Liu et al., 2009; Zhang et al., 2009). We extract GHKM-style rules and restrict that both source and target trees of tree-to-tree rule can contain at most 10 nodes. We further extract SPMT-style full lexicalized rules whose max length of phrase is constrained to 7 on both sides. Same to SPMT, the bilingual phrases are also constrained to be less than 7 words. The final translation quality is evaluated in terms of case-insensitive BLEU-4 with shortest length penalty. The statistical significant test is performed using re-sampling approach (Koehn, 2004).

4.2 Experimental Results and Analysis

We conducted several contrast experiments to demonstrate the effectiveness of bilingual phrase rules. Here we define source phrase rule (SPR for short) as the rule converted from the bilingual phrase whose target side corresponds to a subtree, i.e., we can get a tree-to-tree style rule only by source phrase transformation method. Similarly, target phrase rule (TPR for short) is defined as the rule whose source side corresponds to a subtree.

System	S-trans	T-trans	MT04	MT05
t2t	0	0	30.41	27.68
+TPR	0	2.63	31.12*	28.82*
+SPR	13.08	0	31.90*	28.67*
+SPR+TPR	10.50	2.13	32.10*	28.91*

Table 2. Results (BLEU score) after integrating bilingual phrases. S-trans or T-trans denotes the average number of phrase rules generated by source and target phrase transformation method used in the best translation per sentence in MT04 respectively. The star ‘*’ denotes significantly better than t2t system (p< 0.01).

The results are shown in Table 2. We can clearly see, either TPR or SPR can help to significantly improve the t2t system on the test sets (+0.71 and +1.49 BLEU points for MT04, +1.14 and +0.99 BLEU points for MT05) by effectively exploiting the corresponding bilingual phrases (13.08 source phrase rules and 2.63 target phrase rules on average for the best translation). When we introduce both SPR and TPR into the system, the numbers of the used phrases are slightly decreased, but the BLEU points on MT04 and MT05 continue to go up to 32.10 and 28.91. We have also conducted experiments with all the bilingual phrase rules, but the results are not very stable. We conjecture that this unstable performance is due to the phrase pairs that do not correspond to subtrees on both sides which are syntax unreasonable and might harm the translation quality. Future work will investigate the reason more fully.

System	B-node	U-node	MT04	MT05
t2t	0	2990	30.41	27.68
+SPR+TPR	0	2422	32.10*	28.91*
+TB	3.82(7.94)	2487(49)	33.57*	30.10*
+TB+SPR+TPR	3.12(7.94)	1957(45)	34.50*	31.37*

Table 3: Results (BLEU score) after tree binarization. The meanings of abbreviations are the same with those in Table 2. In addition, TB denotes tree binarization. B-node denotes the average number of new nodes created by binarization used in the best translation on MT04. The numbers in brackets correspond to average number of new nodes created by binarizaiton in each parsing tree. U-node denotes the number of unmatchable nodes in MT04. The numbers in brackets correspond to the number of unmatchable nodes created by binarization.

Table 3 shows the results of systems using binary trees. We can see from the table that only with tree binarization we can significantly improve t2t model by +3.16 and +2.42 BLEU points on MT04 and MT05 respectively. If we further integrate bilingual phrases into the system, the BLEU points on MT04 and MT05 can go up to 34.50 and 31.37, which demonstrates the effectiveness of both bilingual phrases and tree binarization.

Column “B-node” of Table 3 shows the usage of the new nodes created by binarization. With 7.94 new nodes on average for each input parsing tree, almost half of them (3.82 and 3.12 per tree on average) are employed for generating the best translation, which indicates the high efficiency and availability of tree binarization by introducing additional useful translation spans for translation.

Column “U-node” of Table 3 shows the number of *unmatchable nodes* (means none rule can match the subtrees rooted at these nodes) in decoding. By integrating bilingual phrases, the number of unmatchable nodes is reduced from 2990 to 2422. This is the contribution of fuzzy matching method of source phrase rules. With tree binarization, many unmatchable nodes are eliminated, as we can see, from 2990 to 2487, among which only 49 nodes are created by binarization. When we combine the two approaches, the number of unmatchable nodes decreases further (1957 unmatchable nodes), indicating that both bilingual phrases and binarization can help to alleviate the exact matching constraint and enlarge the search space.

4.3 Tree-to-tree vs. State-of-the-art Systems

We also ran Moses (Koehn et al., 2007) with its default settings using the same data and obtained BLEU score of 32.35 and 30.03 on MT04 and MT05 respectively. Our best results are 34.50 and 31.37 on the two test sets which are significant better than Moses.

5 Conclusion and Future Work

To overcome the limitations in rule extraction and decoding procedure of tree-to-tree model, this paper proposed two simple but effective approaches to integrate bilingual phrases and binarize the bilingual parsing trees. The experiments have shown that the approaches yield dramatic improvements over conventional tree-to-tree systems. Furthermore, our improved tree-to-tree model can statistically significantly outperform state-of-the-art phrase-based model Moses.

In future work, we plan to investigate the reasons why the results are unstable after integrating all bilingual phrases. We also plan to use more information to guide the binarization process as the head-out binarization binarizes trees only based on the headword, which is too arbitrary for translation.

Acknowledgments

The research work has been funded by the Natural Science Foundation of China under Grant No. 6097 5053, 61003160 and 60736014 and partially supported by the China-Singapore Institute of Digital Media (CSIDM) project under grant No. CSIDM-200804 as well. We also thank the anonymous reviewers for their valuable suggestions.

References

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

David Chiang, Kevin Knight and Wei Wang, 2009. 11,001 new features for statistical machine translation. In *Proc. of NAACL 2009*, pages 218-226.

David Chiang, 2010. Learning to translate with source and target syntax. In *Proc. of ACL 2010*.

Brooke Cowan, Ivona Kucerova and Michael Collins, 2006. A discriminative model for tree-to-tree translation. In *Proc. of EMNLP*, pages 232-241.

S. DeNeefe, K. Knight, W. Wang, and D. Marcu. 2007. What can syntax-based mt learn from phrase-based mt? In *Proc. of EMNLP2007*.

Yuan Ding and Martha Palmer, 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proc. of ACL 2005*.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. of ACL 2003*.

Michel Galley, Mark Hopkins, Kevin Knight and Daniel Marcu, 2004. What's in a translation rule. In *Proc. of HLT-NAACL 2004*, pages 273–280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang and Ignacio Thayer, 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. of ACL-COLING 2006*.

Liang Huang and David Chiang, 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. of ACL 2007*, pages 144-151.

Liang Huang, Kevin Knight and Aravind Joshi, 2006. A syntax-directed translator with extended domain of locality. In *Proc. of AMTA 2006*, pages 65-73.

Zhongqiang Huang, Martin Cmejrek and Bowen Zhou, 2010. Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions. In *Proc. of EMNLP 2010*, pages 138-147.

P Koehn, H Hoang, A Birch, C Callison-Burch, M Federico, N Bertoldi, B Cowan, W Shen, C Moran and R Zens, 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL 2007*.

Philipp Koehn, 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP 2004*, pages 388–395.

Yang Liu, Qun Liu and Shouxun Lin, 2006. Tree-to-string alignment template for statistical machine translation. In *Proc. of ACL-COLING 2006*.

Yang Liu, Yajuan Lv and Qun Liu, 2009. Improving tree-to-tree translation with packed forests. In *Proc. of ACL-IJCNLP 2009*, pages 558-566.

Daniel Marcu, Wei Wang, Abdessamad Echihabi and Kevin Knight, 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proc. of EMNLP 2006*, pages 44-52.

Haitao Mi, Liang Huang and Qun Liu, 2008. Forest-based translation. In *Proc. of ACL-08*.

Franz Josef Och, 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL 2003*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proc. of ACL 2002*.

Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein, 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of COLING-ACL 2006*, pages 433-440.

Chris Quirk, Arul Menezes and Colin Cherry, 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proc. of ACL 2005*.

Libin Shen, Jinxi Xu and Ralph Weischedel, 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. of ACL-08: HLT*, pages 577-585.

Honglin Sun, Daniel Jurafsky, 2004. Synchronous binarization for machine translation. In *Proc. of HLT-NAACL 2004*.

Wei Wang, Kevin Knight, and Daniel Marcu. 2007. Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proc. of the EMNLP 2007*.

Wei Wang, Jonathan May, Kevin Knight, and Daniel Marcu, 2010. Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Computational Linguistics*, 2010.

Deyi Xiong, Min Zhang, and Haizhou Li. 2010. Learning translation boundaries for phrase-based decoding. *In Proc. of NAACL 2010*, page 136–144.

Hao Zhang, Licheng Fang, Peng Xu, XiaoyunWu, 2011a. Binarized Forest to String Translation. In *Proc. of ACL 2011*.

Hui Zhang, Min Zhang, Haizhou Li, and Chew Lim Tan.2009. Fast Translation Rule Matching for Syntax-based Statistical Machine Translation. In *Proc. of EMNLP 2009*. pages 1037-1045.

Jiajun Zhang, Feifei Zhai, Chengqing Zong, 2011b. Augmenting String-to-Tree Translation Models with Fuzzy Use of Source-side Syntax. In *Proc. of EMNLP 2011*.

Min Zhang, Hongfei Jiang, Ai Ti Aw, Jun Sun, Chew Lim Tan and Sheng Li. 2007. A Tree-to-Tree Alignment- based model for statistical Machine translation. *MT-Summit-07*. 535-542

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan and Sheng Li, 2008. A Tree Sequence Alignment-based Tree-to-Tree Translation Model. In *Proc. of ACL 2008*, pages 559-567.

Andreas Zollmann and Ashish Venugopal, 2006. Syntax augmented machine translation via chart parsing. In *Proc. of Workshop on Statistical Machine Translation 2006*, pages 138-141.