Re-training Monolingual Parser Bilingually for Syntactic SMT

[†]Shujie Liu[*], [‡]Chi-Ho Li, [‡]**Mu Li** and [‡]**Ming Zhou**
[†]School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
shujieliu@mtlab.hit.edu.cn
[‡]Microsoft Research Asia, Beijing, China
{chl, muli, mingzhou}@microsoft.com

Abstract

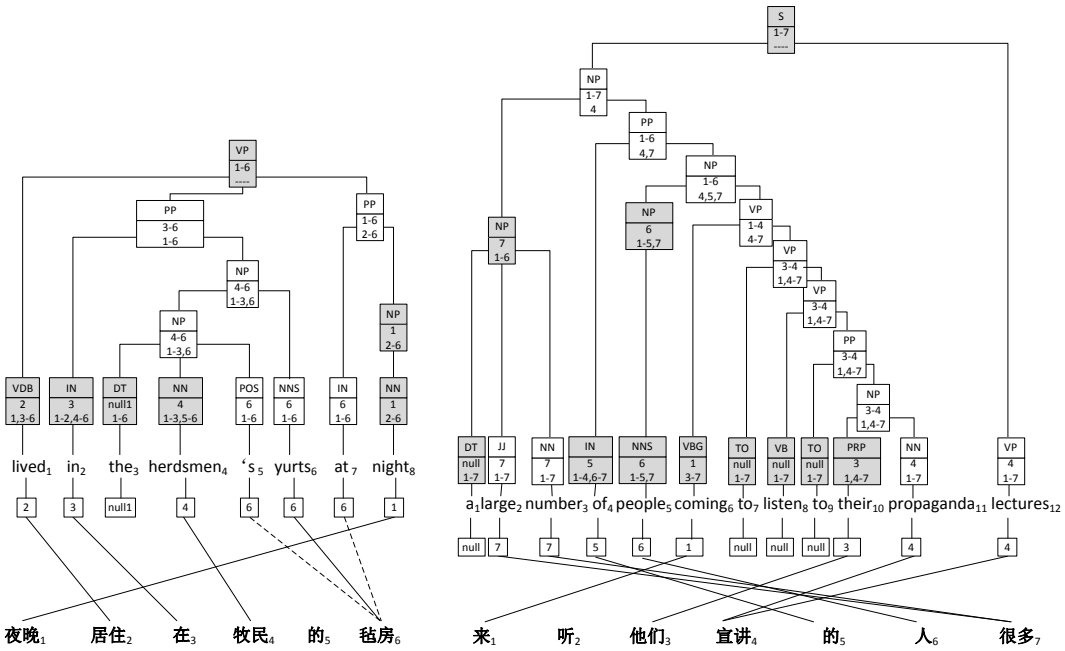
The training of most syntactic SMT approaches involves two essential components, word alignment and monolingual parser. In the current state of the art these two components are mutually independent, thus causing problems like lack of rule generalization, and violation of syntactic correspondence in translation rules. In this paper, we propose two ways of re-training monolingual parser with the target of maximizing the consistency between parse trees and alignment matrices. One is targeted self-training with a simple evaluation function; the other is based on training data selection from forced alignment of bilingual data. We also propose an auxiliary method for boosting alignment quality, by symmetrizing alignment matrices with respect to parse trees. The best combination of these novel methods achieves 3 Bleu point gain in an IWSLT task and more than 1 Bleu point gain in NIST tasks.

1 Introduction

There are many varieties in syntactic statistical machine translation (SSMT). Apart from a few attempts to use synchronous parsing to produce the tree structure of both source language (SL) and target language (TL) simultaneously, most SSMT approaches make use of monolingual parser to produce the parse tree(s) of the SL and/or TL sentences, and then link up the information of the two languages through word alignment. In the current state of the art, word aligner and monolingual parser are trained and applied separately. On the one hand, an average word aligner does not consider the syntax information of both languages, and the output links may violate syntactic correspondence. That is, some SL words yielded by a SL parse tree node may not be traced to, via alignment links, some TL words with legitimate syntactic structure. On the other hand, parser design is a monolingual activity and its impact on MT is not well studied (Ambati, 2008). Many good translation rules may thus be filtered by a good monolingual parser.

In this paper we will focus on the translation task from Chinese to English, and the string-to-tree SSMT model as elaborated in (Galley et al., 2006). There are two kinds of translation rules in this model, minimal rules, and composed rules, which are composition of minimal rules. The minimal rules are extracted from a special kind of nodes, known as frontier nodes, on TL parse tree. The concept of frontier node can be illustrated by Figure 1, which shows two partial bilingual sentences with the corresponding TL sub-trees and word alignment links. The TL words yielded by a TL parse node can be traced to the corresponding SL words through alignment links. In the diagram, each parse node is represented by a rectangle, showing the phrase label, span, and complement span respectively. The span of a TL node N is defined as the minimal contiguous SL string that covers all the SL words reachable from N. The complement span of N is the union of spans of all the nodes that are neither descendants nor ancestors of N (c.f. Galley et al., 2006) . A frontier node is a node of which the span and the complement span do not overlap with each other. In the diagram, frontier nodes are grey in color. Frontier node is the key in the SSMT model, as it identifies the bilingual information which is consistent with both the parse tree and alignment matrix.

There are two major problems in the SSMT model. The first one is the violation of syntactic



(a) (b)

Figure 1. Two example partial bilingual sentences with word alignment and syntactic tree for the target sentence. All the nodes in gray are frontier nodes. Example (a) contains two error links (in dash line), and the syntactic tree for the target sentence of example (b) is wrong.

structure by incorrect alignment links, as shown by the two dashed links in Figure 1(a). These two incorrect links hinder the extraction of a good minimal rule "毡房→ NNS(yurts) " and that of a good composed rule "牧民, 的 → NP(DT(the), NN(herdsmen), POS('s)) ". By and large, incorrect alignment links lead to translation rules that are large in size, few in number, and poor in generalization ability (Fossum et al, 2008). The second problem is parsing error, as shown in Figure 1(b). The incorrect POS tagging of the word "lectures" causes a series of parsing errors, including the absence of the noun phrase "NP(NN(propaganda), NN(lectures))". These parsing errors hinder the extraction of good rules, such as " 宣 讲 → NP(NN(propaganda), NN(lectures)) ".

Note that in Figure 1(a), the parse tree is correct, and the incorrect alignment links might be fixed if the aligner takes the parse tree into consideration. Similarly, in Figure 1(b) some parsing errors might be fixed if the parser takes into consideration the correct alignment links about "propaganda" and "lecture". That is, alignment errors and parsing might be fixed if word aligner and parser are not mutually independent.

In this paper, we emphasize more on the correction of parsing errors by exploiting alignment information. The general approach is to re-train a parser with parse trees which are the most consistent with alignment matrices. Our first strategy is to apply the idea of targeted self-training (Katz-Brown et al., 2011) with the simple evaluation function of frontier set size. That is to re-train the parser with the parse trees which give rise to the largest number of frontier nodes. The second strategy is to apply forced alignment (Wuebker et al., 2010) to bilingual data and select the parse trees generated by our SSMT system for re-training the parser. Besides, although we do not invent a new word aligner exploiting syntactic information, we propose a new method to symmetrize the alignment matrices of two directions by taking parse tree into consideration.

2 Parser Re-training Strategies

Most monolingual parsers used in SSMT are trained upon certain tree bank. That is, a parser is trained with the target of maximizing the agreement between its decision on syntactic structure and that decision in the human-annotated parse trees. As mentioned in Section 1, monolingual syntactic structure is not necessarily suitable for translation, and sometimes the bilingual information in word alignment may help the parser find out the correct structure. Therefore, it is desirable if there is a way to re-train a parser with bilingual information.

What is needed includes a framework of parser re-training, and a data selection strategy that maximizes the consistency between parse tree and alignment matrix. Our two solutions will be introduced in the next two subsections respectively.

2.1 Targeted Self-Training with Frontier Set Based Evaluation (TST-FS)

The first solution is based on targeted self-training (TST) (Katz-Brown et al., 2011). In standard self-training, the top one parse trees produced by the current parser are taken as training data for the next round, and the training objective is still the correctness of monolingual syntactic structure. In targeted self-training, the training objective shifts to certain external evaluation function. For each sentence, the n-best parse trees from the current parser are re-ranked in accordance with this external evaluation function, and the top one of the re-ranked candidates is then selected as training data for the next round. The key of targeted self-training is the definition of this external evaluation function.

As shown by the example in Figure 1(b), an incorrect parse tree is likely to hinder the extraction of good translation rules, because the number of frontier nodes in the incorrect tree is in general smaller than that in the correct tree. Consider the example in Figure 2, which is about the same partial bilingual sentence as in Figure 1(b). Although both parse trees do not have the correct syntactic structure, the tree in Figure 2 has more frontier nodes, leads to more valid translation rules, and is therefore more preferable.

This example suggests a very simple external evaluation function, viz. the size of frontier set. Given a bilingual sentence, its alignment matrix, and the N-best parse trees of the TL sentence, we will calculate the number of frontier nodes for each parse tree, and re-rank the parse trees in its descending order. The new top one parse tree is selected as the training data for the next round of targeted self-training of the TL parser. In the following we will call this approach as targeted self-training with frontier set based evaluation (TST-FS).

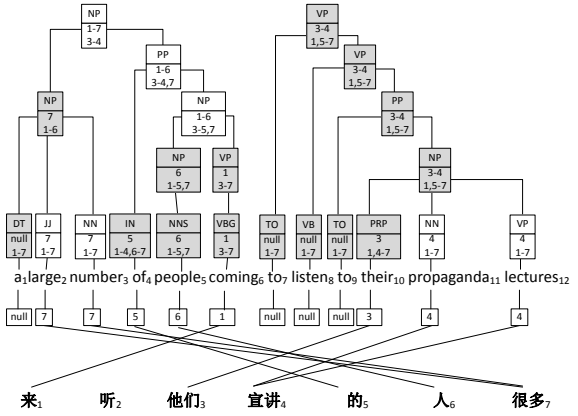


Figure 2. The parse tree selected by TST-FS for the example in Figure 1(b)

Note that the size of the N-best list should be kept small. It is because sometimes a parse tree with an extremely mistaken structure happens to have perfect match with the alignment matrix, thereby giving rise to nearest the largest frontier set size. It is empirically found that a 5-best list of parse trees is already sufficient to significantly improve translation performance.

2.2 Forced Alignment-based Parser Re-Training (FA-PR)

If we doubt that the parse tree from a monolingual parser is not appropriate enough for translation purpose, then it seems reasonable to consider using the parse tree produced by an SSMT system to re-train the parser. A naïve idea is simply to run an SSMT system over some SL sentences and retrieve the by-product TL parse trees for re-training the monolingual parser. The biggest problem of this naïve approach is that the translation by an MT system is often a 'weird' TL sentence, and thus the associated parse tree is of little use in improving the parser.

Forced alignment (Wuebker et al., 2010) of bilingual data is a much more promising approach.

When applied to SSMT, given a bilingual sentence, it performs phrase segmentation of the SL side, parsing of the TL side, and word alignment of the bilingual sentence, using the full translation system as in decoding. It finds the best decoding path that generates the TL side of the bilingual sentence, and the parse tree of the TL sentence is also obtained as a by-product. The parse trees from forced alignment are suitable for re-training the monolingual parser.

Here is the simple iterative re-training algorithm. First we have a baseline monolingual parser and plug it into an SSMT system. Then perform forced alignment, using the SSMT system, of some bilingual data and obtain the parse trees as new training data for the parser. The new parser can then be applied again to do the second round of forced alignment. This iteration of forced alignment followed by parser re-training is kept going until some stopping criterion is met. In the following we will call this approach as forced alignment based parser re-training (FA-PR).

Algorithm 1 Forced Alignment Based Parser Re-Training (FA-PR)

- step1: $t = 0; Pars_0 = Pars_{init}$.
- step2: Use parser $Pars_t$ to parse target sentences of training data, and build a SSMT systems SYS_t.
- step3: Perform forced alignment on training data with SYS_t to get parse trees $Trees_{FAPR}$ for target sentence of training data.
- step4: Train a new parser $Pars_{FAPR}$ with $Trees_{FAPR}$.
- step5: $t = t + 1; Pars_t = Pars_{FAPR}$.
- Step6: Go to step 2, until performance of SYS_t on development data drops, or a preset limit is reached.

There are a few important implementation details of FA-PR. Forced alignment is guaranteed to obtain a parse tree if all translation rules are kept and no pruning is performed during decoding. Yet in reality an average MT system applies pruning during translation model training and decoding, and a lot of translation rules will then be discarded. In order to have more parse trees be considered by forced alignment, we keep all translation rules and relax pruning constraints in the decoder, viz. enlarge the stack size of each cell in the chart from 50 to 150.

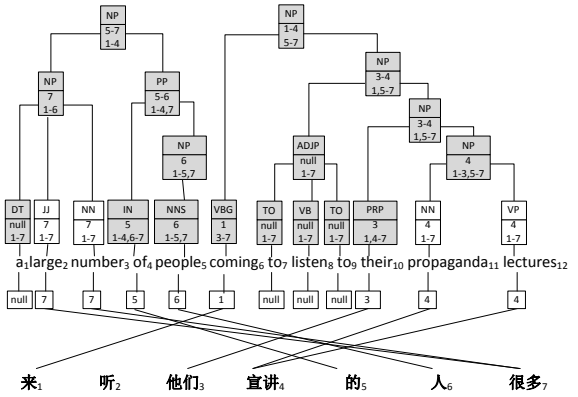


Figure 3. The parse tree selected by FA-PR for the example in Figure 1(b)

Another measure to guarantee the existence of a decoding path in forced alignment is to allow part of a SL or TL sentence translate to null. Consider the example in Figure 1(b). We also add a null alignment for any span of the source and target sentences to handle the null translation scenario. It is easy to add a null translation candidate for a span of the source sentence during decoding, but not easy for target spans. For example, suppose the best translation candidate for the source span " 来 1 NP 的 5 人 6 很多 7" is "a large number of people coming NP", and the best translation candidate for "听 2 他们 3 宣讲 4" is "their propaganda lectures", there is no combination of candidates from two n-best translation lists which can match a sequence in the given target part, so we add a translation candidate ("to listen to ") generated from null, whose syntactic label can be any label (decided according to the translated context, which is "ADJP" here). The feature weights for the added null alignment are set to be very small, so as to avoid the competition with the normal candidates. In order to generate normal trees with not so many null alignment sub-trees for the target sentence (such trees are not suitable for parser re-training), only target spans with less than 4 words can align to null, and such null-aligned sub-tree can only be added no more than 3 times.

With all the mentioned modification of the forced alignment, the partial target tree generated using forced alignment for the example in Figure 1(b) is shown in Figure 3. We can see that even

with an incorrect sub-tree, more useful rules can be extracted, compared with the baseline sub-tree and the sub-tree generated from TST-FS.

3 Word Alignment Symmetrization

The most widely used word aligners in MT, like HMM and IBM Models (Och and Ney, 2003), are directional aligners. Such aligner produces one set of alignment matrices for the SL-to-TL direction and another set for the TL-to-SL direction. Symmetrization refers to the combination of these two sets of alignment matrices.

The most popular method of symmetrization is intersect-diag-grow (IDG). Given a bilingual sentence and its two alignment matrices A_{ST} and $A_{TS,}$ IDG starts with all the links in $A_{ST} \cap A_{TS}$. Then IDG considers each link in $A_{ST} \cup A_{TS} - (A_{ST} \cap A_{TS})$ in turn. A link is added if its addition does not make some phrase pairs overlap. Although IDG is simple and efficient, and has been shown to be effective in phrase-based SMT, it is problematic in SSMT, as illustrated by the example in section 1.

3.1 Intersect-Diag-Syntactic-Grow (IDSG)

We propose a new symmetrization method, Intersect-Diag-Syntactic-Grow (IDSG), which is an adaptation of IDG but also taking syntactic information in consideration. It is sketched in Algorithm 2.

Algorithm 2 Intersect-Diag-Syntactic-Grow

- step1: Generate all the candidate links A_{candi} using IDG.
- step2: Select the one which can generate the biggest frontier set:

$$l = \underset{l' \in A_{candi}}{\operatorname{argmax}}(frontierSize(A \cup l', Tree))$$

- step3: Add l to A, and repeat step 1, until no new link can be added.

Like IDG, IDSG starts with all the links in $A_{ST} \cap A_{TS}$ and its main task is to add links selected from $A_{candi} = A_{ST} \cup A_{TS} - (A_{ST} \cap A_{TS})$. IDSG is also subject to the constraints of IDG. The new criterion in link selection in IDSG is specified in Step 2. Given a parse tree of the TL side of the bilingual sentence, in each iteration IDSG considers the change of frontier set size caused by the addition of each link in A_{candi}. The link leading to the maximum number of frontier nodes is added (and removed from A_{candi}). This process continues until no more links can be added.

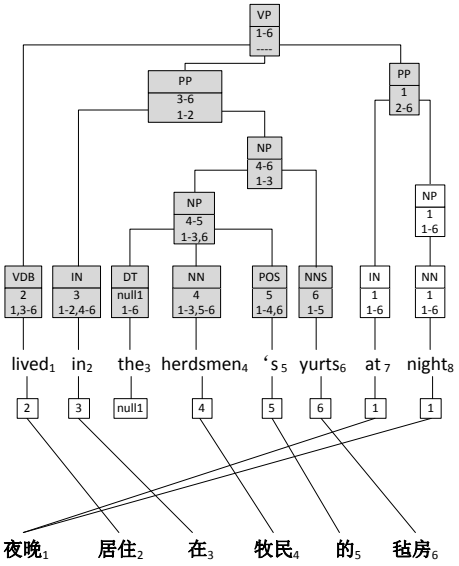


Figure 4, the alignment generated by IDSG for the example in Figure 1(a)

In sum, IDSG add links in an order which take syntactic structure into consideration, and the link with the least violation of the syntactic structure is added first.

For the example in Figure 1(a), IDSG succeeds in discarding the two incorrect links, and produces the final alignment and frontier set as shown in Figure 4. Note that IDSG still fails to produce the correct link (the$_3$, 牧民$_4$), since this link does not appear in A_{candi} at all.

3.2 Combining TST-FS/FA-PR and IDSG

Parser re-training aims to improve a parser with alignment matrix while IDSG aims to improve alignment matrix with parse tree. It is reasonable to combine them, and there are two alternatives of the combination, depending on the order of application. That is, we could either improve alignment matrix by IDSG and then re-train parser with the better alignment, or re-train parser and then improve alignment matrix with better syntactic information. Either alternative can be arranged into an iterative training routine, but empirically it is found that only one round of parser re-training before or after only one round of IDSG is already enough.

4 Experiment

In this section, we conduct experiments on Chinese to English translation task to test our proposed methods of parser re-training and word alignment symmetrization. The evaluation method is the case insensitive IBM BLEU-4 (Papineni et al., 2002). Significant testing is carried out using bootstrap re-sampling method proposed by Koehn (2004) with a 95% confidence level.

4.1 Parser and SMT Decoder

The syntactic parser we used in this paper is Berkley parser, with the grammar trained on WSJ corpus, and the training method follows Petrov and Klein (2007). Our SMT decoder is an in-house implementation of string-to-tree decoder. The features we used are standard used features, such as translation probabilities, lexical weights, language model probabilities and distortion probability. The feature weights are tuned using the minimum error rate training (MERT) (Och, 2003).

4.2 Experiment Data Setting and Baselines

We test our method with two data settings: one is IWSLT data set, the other is NIST data set.

	dev8+dialog	dev9
Baseline	50.58	49.85

Table 1. Baselines for IWSLT data set

	NIST'03	NIST'05	NIST'08
Baseline	37.57	36.44	24.87

Table 2. Baselines for NIST data set

Our IWSLT data is the IWSLT 2009 dialog task data set. The training data include the BTEC and SLDB training data. The training data contains 81k sentence pairs, 655k Chinese words and 806k English words. The language model is 5-gram language model trained with the English sentences in the training data. We use the combination of dev8 and dialog as development set, and dev9 as test set. The TL sentences of the training data with the selected/generated trees are used as the training data to re-train the parser. To get the baseline of this setting, we run IDG to combine the bi-direction alignment generated by Giza++ (Och Ney, 2003), and run Berkeley parser (Petrov and Klein, 2007) to parse the target sentences. With the baseline alignments and syntactic trees, we extract rules and calculate features. The baseline results are shown in Table 1.

For the NIST data set, the bilingual training data we used is NIST 2008 training set excluding the Hong Kong Law and Hong Kong Hansard. The training data contains 354k sentence pairs, 8M Chinese words and 10M English words, and is also the training data for our parser re-training. The language model is 5-gram language model trained with the Giga-Word corpus plus the English sentences in the training data. The development data to tune the feature weights of our decoder is NIST 2003 evaluation set, and test sets are NIST 2005 and 2008 evaluation sets. The baseline for NIST data is got in a similar way with for IWSLT, which are shown in Table 2 .

4.3 Results of TST-FS/ FA-PR

The parser re-training strategies TST-FS and FA-PR are tested with two baselines, one is the default parser without any re-training and another is standard self-training (SST). All three re-training approaches are based on the same bilingual datasets as used in translation model training. The MT performances on IWSLT and NIST by the four approaches are shown in Table 3 and 4 respectively.

It can be seen that just standard self-training does improve translation performance, as re-training on the TL side of bilingual data is a kind of domain adaptation (from WSJ to IWSLT/NIST). But targeted self-training achieves more noticeable improvement, almost twice as much as standard self-training. This confirms the value of word alignment information in parser re-training. Finally, the even larger improvement of FA-PR than TST-FS shows that merely increasing the number of frontier nodes is not enough. Some frontier nodes are of poor quality, and the frontier nodes found in forced alignment are more suitable.

It can also be seen that the improvement in IWSLT is larger than that in NIST. The first reason is that both WSJ and NIST are of the news domain and of formal writing style, whereas IWSLT is of the tourist domain and of colloquial style. Therefore any improvement from the default parser, which is trained on WSJ, is expected to be smaller in the NIST case. Another reason is that, since the

IWSLT dataset is much smaller, the impact of more and better rules is more obvious.

Note that the figures in Table 3 and 4 are about parser re-training for only one iteration. It is found that, more iteration do not lead to further significant improvement. The forced alignment of bilingual training data does not obtain a full decoding path for every bilingual sentence. It is because, although all translation rules are kept, there is still pruning during decoding. Only 64% of the IWSLT dataset and 53% of the NIST dataset can be successfully forced-aligned. In general, the longer the bilingual sentence, the less likely forced alignment is successful, and that is why a lower proportion of NIST can be forced-aligned.

4.4 Symmetrization

The new symmetrization method IDSG is compared with the baseline method IDG.

	dev8+dialog	dev9	# Rules
IDG	50.58	49.85	515K
IDSG	**52.71** (+2.31)	**51.80** (+2.05)	626K

Table 5. MT performance of symmetrization methods on IWSLT data set. The results in bold type are significantly better than the performance of IDG.

	NIST'03	NIST'05	NIST'08	#Rules
IDG	37.57	36.44	24.87	3,376K
IDSG	38.15 (+0.58)	**37.07** (+0.63)	**25.67** (+0.80)	4,109K

Table 6. MT performance of symmetrization methods on NIST data. The results in bold type are significantly better than the performance of IDG.

As shown by the results in Table 5 and 6, IDSG enlarges the set of translation rules by more than 20%, thereby improving translation performance significantly. As in parser re-training, the improvement in the IWSLT task is larger than that in the NIST task. Again, it is because the IWSLT dataset is very small and so the effect of rule table size is more obvious.

	dev8+dialog	dev9	# Rules
Baseline	50.58	49.85	515K
SST	**52.04** (+1.46)	**51.26** (+1.41)	574K
TST-FS	**52.75** (+2.17)	**52.51** (+2.66)	572K
FA-PR	**53.31** (+2.73)	**52.8** (+2.95)	591K

Table 3. MT performance of parser re-training strategies on IWSLT data set. The results in bold type are significantly better than the baseline.

	NIST'03	NIST'05	NIST'08	#Rules
Baseline	37.57	36.44	24.87	3,376K
SST	37.98 (+0.41)	36.79 (+0.35)	25.30 (+0.43)	3,462K
TST-FS	**38.42** (+0.85)	**37.39** (+0.95)	**25.79** (+0.92)	3,642K
FA-PR	**38.74** (+1.17)	**37.69** (+1.25)	**25.89** (+1.02)	3,976K

Table 4. MT performance of parser re-training strategies on NIST data set. The results in bold type are significantly better than the baseline.

4.5 Methods combined

As mentioned in section 3.2, parser re-training and the new symmetrization method can be combined in two different ways, depending on the order of application. Table 7 and 8 show the experiment results of combining FA-PR with IDSG.

It can be seen that either way of the combination is better than using FA-PR or IDSG alone. Yet there is no significant difference between the two kinds of combination.

The best result is a gain of more than 3 Bleu points on IWSLT and that of more than 1 Bleu point on NIST.

5 Related Works

There are a lot of attempts in improving word alignment with syntactic information (Cherry and Lin, 2006; DeNero and Klein, 2007; Hermjackob, 2009) and in improving parser with alignment information (Burkett and Klein, 2008). Yet strictly speaking all these attempts aim to improve the

parser/aligner itself rather than the translation model.

To improve the performance of syntactic machine translation, Huang and Knight (2006) proposed a method incorporating a handful of relabeling strategies to modify the syntactic trees structures. Ambati and Lavie (2008) restructured target parse trees to generate highly isomorphic target trees that preserve the syntactic boundaries of constituents aligned in the original parse trees. Wang et al., (2010) proposed to use re-structuring and re-labeling to modify the parser tree. The re-structuring method uses a binarization method to enable the reuse of sub-constituent structures, and the linguistic and statistical re-labeling methods to handle the coarse nonterminal problem, so as to enhance generalization ability. Different from the previous work of modifying tree structures with post-processing methods, our methods try to learn a suitable grammar for string-to-tree SMT models, and directly produce trees which are consistent with word alignment matrices.

Instead of modifying the parse tree to improve machine translation performance, many methods were proposed to modify word alignment by taking syntactic tree into consideration, including deleting incorrect word alignment links by a discriminative model (Fossum et al., 2008), re-aligning sentence pairs using EM method with the rules extracted with initial alignment (Wang et al., 2010), and removing ambiguous alignment of functional words with constraint from chunk-level information during rule extraction (Wu et al., 2011). Unlike all these pursuits, to generate a consistent word alignment, our method modifies the popularly used IDG symmetrization method to make it suitable for string-to-tree rule extraction, and our method is much simpler and faster than the previous works.

6 Conclusion

In this paper we have attempted to improve SSMT by reducing the errors introduced by the mutual independence between monolingual parser and word aligner. Our major contribution is the strategies of re-training parser with the bilingual information in alignment matrices. Either of our proposals of targeted self-training with frontier set size as evaluation function and forced alignment based re-training is more effective than baseline parser or standard self-training of parser. As an auxiliary method, we also attempted to improve alignment matrices by a new symmetrization method.

In future, we will explore more alternatives in integrating parsing information and alignment information, such as discriminative word alignment using a lot of features from parser.

	dev8+dialog	dev9	# Rules
Baseline	50.58	49.85	515K
IDSG	**52.71** (+2.31)	**51.80** (+2.05)	626K
FA-PR	**53.31** (+2.73)	**52.8** (+2.95)	591K
IDSG then FA-PR	**53.64** (3.06)	**53.32** (+3.47)	602K
FA-PR then IDSG	**53.81** (+3.23)	**53.26** (+3.41)	597K

Table 7. MT performance of the new methods on IWSLT data set. The results in bold type are significantly better than the baseline.

	NIST'03	NIST'05	NIST'08	#Rules
Baseline	37.57	36.44	24.87	3,376K
IDSG	38.15 (+0.58)	**37.07** (+0.63)	**25.67** (+0.80)	4,109K
FA-PR	**38.74** (+1.17)	**37.69** (+1.25)	**25.89** (+1.02)	3,976K
IDSG then FA-PR	**38.97** (+1.40)	**37.95** (+1.51)	**26.74** (+1.87)	4,557K
FA-PR then IDSG	**38.90** (+1.33)	**37.94** (+1.50)	**26.52** (+1.65)	4,478K

Table 8. MT performance of the new methods on NIST data set. The results in bold type are significantly better than the baseline.

References

Vamshi Ambati and Alon Lavie. 2008. Improving syntax driven translation models by re-structuring divergent and non-isomorphic parse tree structures. In *Student Research Workshop of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 235-244.

David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 877-886.

Colin Cherry and Dekang Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

John DeNero and Dan Klein. 2007. Tailing word alignment to syntactic machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 17-24.

Victoria Fossum, Kevin Knight, Steven Abney. 2008. Using syntax to improve word alignment precision for syntax-based machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 44-52.

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 *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961-968.

Ulf Hermjackob. Improved word alignment with statistics and linguistic heuristics. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 229-237.

Bryant Huang, Kevin Knight. 2006. Relabeling syntax trees to improve syntax-based machine translation quality. In *Proceedings of the Human Technology Conference of the North American Chapter of the ACL*, pages 240-247.

Jason Katz-Brown, Slav Petrov, Ryan McDonald, Franz Och, David Talbot, Hiroshi Ichikawa, Masakazu Seno, Hideto Kazawa. 2011. Training a parser for machine translation reordering. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 183-192.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 388-395.

Wei Wang, Jonathan May, Kevin Knight, Daniel Marcu. 2010. Re-structuring, re-labeling, and re-alignment for syntax-Based machine translation. *Computational Linguistics*, 36(2).

Xianchao Wu, Takuya Matsuzaki and Jun'ichi Tsujii. 2011. Effective use of function words for rule generalization in forest-based translation. In *Proceedings of the Association for Computational Linguistics*, pages 22-31.

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

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

Joern Wuebker, Arne Mauser and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the Association for Computational Linguistics*, pages 475-484.

Kishore Papineni, Salim Roukos, Todd Ward and Wei-jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 311-318.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In Proceedings of *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 404–411.