Parser Self-Training for Syntax-Based Machine Translation

Makoto Morishita, Koichi Akabe, Yuto Hatakoshi
Graham Neubig, Koichiro Yoshino, Satoshi Nakamrua

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara, Japan
{morishita.makoto.mb1,akabe.koichi.zx8,neubig,koichiro,s-nakamura}@is.naist.jp
hatakoshi.yuto@gmail.com

Abstract

In syntax-based machine translation, it is known that the accuracy of parsing greatly affects the translation accuracy. Self-training, which uses parser output as training data, is one method to improve the parser accuracy. However, because parsing errors cause noisy data to be mixed with the training data, automatically generated parse trees do not always contribute to improving accuracy. In this paper, we propose a method for selecting self-training data by performing syntax-based machine translation using a variety of parse trees, using automatic evaluation metrics to select which translation is better, and using that translation's parse tree for parser self-training. This method allows us to automatically choose the trees that contribute to improving translation accuracy, improving the effectiveness of self-training. In experiments, we found that our self-trained parsers significantly improve a state-of-the-art syntax-based machine translation system in two language pairs.

1. Introduction

In statistical machine translation (SMT), representative methods include Phrase-Based Machine Translation (PBMT) [1], in which each phrase is translated by the translation model and reordered to the appropriate target language order, and syntax-based machine translation [2], which uses parts of syntactic parse trees for translation. While PBMT generally achieves high accuracy on language pairs with close word order such as English-French, syntax-based machine translation techniques have shown to allow for better translation accuracy on language pairs with different word order such as English-Japanese.

Among the various methods for syntax-based translation, Tree-to-String (T2S) translation [3], which uses parse trees in the source language, has been reported to achieve high translation accuracy while maintaining translation speed [4]. However, T2S translation uses parser results in the source language, so translation accuracy greatly depends on parser accuracy. One method to ameliorate this problem is Forest-to-String (F2S) translation [5], which considers multiple parse trees during the decoding process. Even F2S translation, however, is heavily affected by the accuracy of the parser used to generate the parse forest [4].

Parser *self-training* is one method to improve parser accuracy [6]. Self-training first parses unannotated sentences using an existing model, then uses these automatically generated parse trees to retrain the parser. This allows the parser to automatically adapt to the data used for self-training, increasing coverage of vocabulary or syntactic structures, and thus increasing parser accuracy. However, one downside of standard self-training methods is that automatically generated parse trees are often incorrect, which reduces their effectiveness as training data.

While there has not been much work on self-training in the context of syntax-based machine translation itself, Katz-Brown et al. [7] have proposed a method for self-training in the context of syntactic pre-ordering. In this method, which they call *targeted self-training*, they first generate multiple parse trees using a syntactic parser, then use these trees to perform pre-ordering, score the output by comparing to correct pre-ordered data, and select the parse tree that has the highest score. By using information about the correct pre-ordering to select which parse tree to use, this method has the ability to remove parse trees that result in incorrect pre-orderings, reducing noise in the training data. However, on the down side, making the manually aligned data required to apply this variety of targeted self-training is costly, limiting its applicability to situations where this data can be created.

In this paper, we propose a method for targeted self-training of parsers for syntax-based translation. The proposed method is applicable not only to pre-ordering but also syntax-based MT, and has the additional advantage that it does not require the preparation of costly hand-aligned training data because it chooses data using standard MT automatic evaluation metrics. This allows for the use of existing bilingual corpora as training data for targeted self-training, making it possible to improve parsers and F2S translation accuracy in a wider variety of fields. By carrying out experiments on targeted self-training considering machine translation accuracy, we confirmed that the proposed method significantly improves the translation accuracy of a state-of-the-art F2S system in two language pairs.

2. Tree-to-String translation

In SMT, given the source sentence $\boldsymbol{f}$, we consider the problem of finding translation $\hat{e}$ that maximizes the posterior probability $Pr(\boldsymbol{e}|\boldsymbol{f})$

$$\hat{\boldsymbol{e}} := \operatorname*{argmax}_{\boldsymbol{e}} Pr(\boldsymbol{e}|\boldsymbol{f}). \tag{1}$$

Among the varieties of SMT, T2S translation uses source language parse tree $T_{\boldsymbol{f}}$ to disambiguate the source structure and express the hierarchical relationships between the source and target languages as rules, allowing for more accurate translation. T2S translation can be formulated as follows

$$\begin{aligned}\hat{\boldsymbol{e}} &:= \operatorname*{argmax}_{\boldsymbol{e}} Pr(\boldsymbol{e}|\boldsymbol{f}) & (2)\\ &= \operatorname*{argmax}_{\boldsymbol{e}} \sum_{T_{\boldsymbol{f}}} Pr(\boldsymbol{e}|\boldsymbol{f}, T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) & (3)\\ &\simeq \operatorname*{argmax}_{\boldsymbol{e}} \sum_{T_{\boldsymbol{f}}} Pr(\boldsymbol{e}|T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) & (4)\\ &\simeq \operatorname*{argmax}_{\boldsymbol{e}} Pr(\boldsymbol{e}|\hat{T}_{\boldsymbol{f}}), & (5)\end{aligned}$$

where $\hat{T}_{\boldsymbol{f}}$ is the highest probability parse tree candidate represented by the following formula:

$$\hat{T}_{\boldsymbol{f}} = \operatorname*{argmax}_{T_{\boldsymbol{f}}} Pr(T_{\boldsymbol{f}}|\boldsymbol{f}). \tag{6}$$

As shown in Figure 1, translation rules used by T2S translation[1] are represented by the set of a source subtree and a target language string of words, including the replaceable variables x. In the example shown in Figure 1, x_0 and x_1 are the replaceable variables. During translation, the decoder finds the highest probability translation considering the probability of translation rules, language models, or other features. The decoder can also be used to output the n translations with the highest probability, n-best translations.

In T2S translation, by taking the source language parse tree into account, translation of long-distance word ordering can be more accurate than PBMT. However, because T2S uses the parse tree for translation, the translation accuracy greatly depends on the parser accuracy. As mentioned in the introduction, F2S translations reduce the adverse effect of parser errors by using a parse forest, which is a hyper-graph efficiently expressing a large number of parse trees. By using a parse forest for translation, the decoder can select which parse tree to use from several parse tree candidates, leading to improved translation accuracy [9]. F2S translation can be formulated as follows:

$$\langle \hat{e}, \hat{T}_{\boldsymbol{f}} \rangle = \operatorname*{argmax}_{\langle \boldsymbol{e}, T_{\boldsymbol{f}} \rangle} Pr(\boldsymbol{e}|T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}). \tag{7}$$

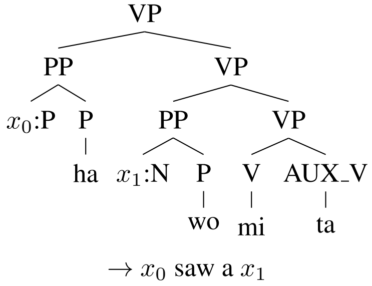


Figure 1: An example of a Japanese-to-English tree-to-string translation rule

However, even using F2S translation, the accuracy is heavily affected by the accuracy of the parser used to generate the parse forest [4]. In the following sections, we describe some methods to improve parser accuracy.

3. Parser self-training

3.1. Introduction of self-training

Parser self-training retrains the parser using the parse trees automatically generated by an existing model, allowing the parser to adapt to the data used for self-training and improve accuracy. In other words, for each sentence used for self-training, we find the highest probability parse tree $\hat{T}_{\boldsymbol{f}}$ based on Equation (6), then use this parse tree to retrain the parser.

When Charniak first proposed parser self-training, he reported that a parser using Probabilistic Context-Free Grammar (PCFG) models trained using the WSJ corpus [10] achieved no gain through self-training [11]. On the other hand, the PCFG with Latent Annotations (PCFG-LA) model, which achieves improved parsing accuracy by using latent annotations, has been reported to be improved significantly by self-training [12]. This is because the PCFG-LA has relatively high accuracy, so the automatically generated parse trees used for self-training are more accurate, and because the PCFG-LA model has more parameters than standard PCFGs, making it more likely to benefit from the increased amount of data. Based on these studies, we consider parser self-training using the PCFG-LA model.

3.2. Self-training of the parser in machine translation

As mentioned in the introduction, there is one previous study on improving translation accuracy by doing parser self-training. Katz-Brown et al. propose a method for *targeted self-training*, where trees are selected based on an extrinsic evaluation measure, and report that it is possible improve translation accuracy itself [7]. Specifically, they automatically generate several candidate parse trees, then select the candidate for which the pre-ordering result is most similar to hand-aligned correct data. This tree is then used to retrain the parser.

Formally, we define the pre-ordering function $\text{reord}(T_{\boldsymbol{f}})$,

[1] Specifically, T2S translation using tree transducers [8].

which generates a pre-ordered source language sentence $\boldsymbol{f}'$ based on a parse tree $T_{\boldsymbol{f}}$, and a score function $\mathrm{score}(\boldsymbol{f}'^{*}, \boldsymbol{f}')$ [13], which compares $\boldsymbol{f}'$to the reference preordered sentence $\boldsymbol{f}'^{*}$. Parse tree $\bar{T}_{\boldsymbol{f}}$, which is used in self-training, is selected from the candidate parse trees $\boldsymbol{T_f}$ by the following formula

$$\bar{T}_{\boldsymbol{f}} = \underset{T_{\boldsymbol{f}} \in \boldsymbol{T_f}}{\mathrm{argmax}}\, \mathrm{score}(\boldsymbol{f}'^{*}, \mathrm{reord}(T_{\boldsymbol{f}})) \tag{8}$$

In this paper, based on these previous studies, we propose a method for parser targeted self-training for syntax-based translation. In the following sections, we explain the detail of this method and verify its effectiveness.

4. Parser self-training for syntax-based MT

An important point that determines the effectiveness of self-training is how to select the data used to retrain the parser. In the following sections, we propose several methods to select parse trees and sentences that are effective to improve the accuracy of F2S translation.

4.1. Selecting parse trees

As described in Section 3.2, the targeted self-training proposed by Katz-Brown et al. [7], selects the most accurate parse tree from n-best candidates by comparing hand-aligned correct preordering data and automatically generated parse trees. However, constructing hand-aligned data is costly, and thus it is impractical to create large data sets for this method. To solve this problem, we propose two methods for targeted self-training using only a parallel corpus. One is to use the parse tree used in the 1-best translation selected by the decoder, and the other is to use the parse tree used in the oracle translation, which is the most similar translation to the reference translation from the n-best list selected by automatic evaluation metrics.

4.1.1. Decoder 1-best

As described in Section 2, in F2S translation, the decoder selects the parse tree used to generate the translation with the highest probability from the parse forest. A previous study has noted that the F2S decoder has the ability to select more accurate parse trees, as it uses other feature functions such as rule probabilities and language model probabilities, which cannot be considered by the baseline parser [9]. Thus, the parse tree used in the one-best translation could be more effective for self-training than the parser 1-best tree. In this case, the parse tree used in self-training is $\hat{T}_{\boldsymbol{f}}$ in Formula (7).

4.1.2. Automatic evaluation 1-best

During translation, the decoder outputs the translation that has the highest translation probability from a multitude of translation candidates. However, there are also cases in which other candidate translations, for example ones in the n-best list, are more similar to the reference translation, which indicates that they may be more accurate than the decoder 1-best translation.

We define the oracle translation $\bar{\boldsymbol{e}}$, which is the closest to reference translation $\boldsymbol{e}^{*}$ among the n-best translation candidates $\boldsymbol{E}$. In this method, we perform self-training using the parse tree that is used in the oracle translation $\bar{\boldsymbol{e}}$. By using the score function $\mathrm{score}(\cdot)$, which represents the similarity between the hypothesis and reference translation, $\bar{\boldsymbol{e}}$ is formulated as follows:

$$\bar{\boldsymbol{e}} = \underset{\boldsymbol{e} \in \boldsymbol{E}}{\mathrm{argmax}}\, \mathrm{score}(\boldsymbol{e}^{*}, \boldsymbol{e}). \tag{9}$$

4.2. Selecting sentences

In Section 4.1, we described methods to select, from an n-best list for a single sentence, parse trees that may be useful in self-training. However, in many cases, the correct parse tree may not be included in the n-best translation candidates, and there is a possibility of these sentences adding noise to the training data. Therefore, we further propose two methods for selecting which sentences should be used in self-training from the entirety of the training data, potentially removing sentences for which no good n-best candidate exists. One is to use sentences for which the translated sentence's automatic evaluation score exceeds a threshold, and the other is to use sentences that have a large score increase between the decoder 1-best and oracle translation.

4.2.1. Automatic evaluation threshold

There are some sentences in the corpus that are not translated accurately by the MT system, and the score of automatic evaluation metrics decreases. The cause of low evaluation scores could be for the the following reasons:

- An incorrect parse tree has been used in the translation.
- The translation model does not sufficiently cover the source sentence's vocabulary or phrases.
- The reference translation, which is used to calculate the automatic evaluation score, is a free or incorrect translation, and the system cannot output a similar translation.

In such cases, the score of automatic evaluation metrics will be low even for the oracle translation. Because the F2S decoder cannot select the correct trees or the evaluation scores are not reliable, these data are more likely to have noisy oracle trees for training, and thus it is potentially beneficial to exclude these trees from the training data. For these reasons, we propose a sentence selection method that uses only sentences that achieve scores over a threshold, which can be expected to have more accurate parse trees. The set of sentences for self-training is defined as follows, where t is the threshold, $\boldsymbol{e}^{*(i)}$ is the reference translation of the sentence i, $\bar{\boldsymbol{e}}^{(i)}$ is the oracle translation of the sentence i, $\bar{\boldsymbol{E}}$ is the set of all oracle translations, and score(e) is the automatic evaluation score function

$$\{i \mid \text{score}(\boldsymbol{e}^{*(i)}, \bar{\boldsymbol{e}}^{(i)}) \geq t,\ \bar{\boldsymbol{e}}^{(i)} \in \bar{\boldsymbol{E}}\}. \tag{10}$$

4.2.2. Automatic evaluation gain

Next, we focused on the difference between the automatic evaluation score of the decoder 1-best and oracle translation. In the case that the parse forest output by the parser has incorrect probabilities for the parse trees, in many cases, the decoder will select the incorrect parse tree and output the wrong translation. On the other hand, the oracle translation is more likely to have used a correct parse tree from the parse forest. Therefore, by using the parse trees used in oracle translations as training data in these cases, it may be possible to improve the parser's probability estimates. This will result in the system using the self-trained parser tending to output the correct translation as a 1-best, improving translation accuracy.

To select the sentences, we define the function gain$(\bar{e}^{(i)}, \hat{e}^{(i)})$, which represents the gain between the score of 1-best translation $\hat{e}^{(i)}$ and oracle translation $\bar{e}^{(i)}$, then selects the sentences with highest gain as in Formula (10). The function gain$(\bar{e}^{(i)}, \hat{e}^{(i)})$ is formulated as follows:

$$\text{gain}(\bar{\boldsymbol{e}}^{(i)}, \hat{\boldsymbol{e}}^{(i)}) = \text{score}(\bar{\boldsymbol{e}}^{(i)}) - \text{score}(\hat{\boldsymbol{e}}^{(i)}). \tag{11}$$

In addition, in this method, in order to ensure that the sentence length distribution of the training data is similar to that of the entire corpus, we use the following formula proposed by Gascó et al. [14], to ensure that the length distribution of the selected sentences is similar to that of the overall corpus distribution.[2] In this formula, $|\boldsymbol{e}|$ is the length of target language sentence $\boldsymbol{e}$, $|\boldsymbol{f}|$ is the length of source language sentence $\boldsymbol{f}$, $N_c(|\boldsymbol{e}| + |\boldsymbol{f}|)$ is the number of sentences in the corpus with length $|\boldsymbol{e}| + |\boldsymbol{f}|$, and N_c is the number of sentences in the entire corpus

$$p(|\boldsymbol{e}| + |\boldsymbol{f}|) = \frac{N_c(|\boldsymbol{e}| + |\boldsymbol{f}|)}{N_c}. \tag{12}$$

The number of sentences selected for the self-training set is formulated as follows, where $N_t(|\boldsymbol{e}| + |\boldsymbol{f}|)$ is the number of sentences in the self-training set with length $|\boldsymbol{e}| + |\boldsymbol{f}|$, and N_t is the number of sentences in the entire self-training set

$$N_t(|\boldsymbol{e}| + |\boldsymbol{f}|) = p(|\boldsymbol{e}| + |\boldsymbol{f}|)N_t. \tag{13}$$

5. Experiments

5.1. Experimental setup

In the experiments, we focused on Japanese-English and Japanese-Chinese translation. Because the amount of hand-labeled Japanese parse tree data is less than that available for English, the Japanese parser is prone to parse errors. As the translation data, we use ASPEC,[3] which is a parallel corpus of scientific papers abstracts. The number of sentences in ASPEC is shown in Table 1.[4] As a state-of-the-art baseline for verifying the effect of self-training, we use the system developed by Neubig [15],[5] which was the most accurate system on the Workshop on Asian Translation (WAT) 2014 [16]. We use Travatar [17] as a Forest-to-String decoder. As a parser, we use the PCFG-LA parser Egret,[6] and train a baseline model on a phrase-structure version of the Japanese Dependency Corpus (JDC) [18], which has about 7000 sentences. Forests were pruned to remove hyper-edges which do not appear in the 100 n-best trees. Egret sometimes fails to output a parse tree, and in this case, we remove the failed sentences. We evaluate the accuracy by using two automatic evaluation metrics, BLEU [19] and RIBES [20], and to evaluate the accuracy for each sentence in sentence or oracle selection, we use BLEU+1 [21]. For self-training data, we add the data selected from the ASPEC training data to the JDC trees. The training data for the translation systems is parsed using the standard JDC model, and the self-trained models are used only to parse the development and test corpora at test time.[7] We verify statistical significance using the bootstrap resampling method [22]. In the next section, we compare the following parser self-training methods:

Table 1: The number of sentences in ASPEC

	Train	Dev	DevTest	Test
Ja-En	2,000,000	1,790	1,784	1,812
Ja-Zh	672,315	2,090	2,148	2,107

Parser 1-best

As in Formula (6), we use the 1-best parse trees for self-training. We select the sentences randomly from the corpus.[8]

MT 1-best

As described in Section 4.1.1, we input the parse forest to the decoder, and use the parse trees used in the 1-best translation. We select the sentences randomly from the corpus as in Parser 1-best.

Oracle

As described in Section 4.1.2, we input the parse forest to the decoder, output unique 500-best hypotheses and use the parse tree corresponding to the translation that has the highest BLEU+1 score in this n-best list. We

[2] The BLEU gain approach was not effective if we did not use this technique, as it tends to select only short sentences where small changes in wording cause large changes in evaluation scores.

[3] http://lotus.kuee.kyoto-u.ac.jp/ASPEC

[4] ASPEC actually has 3.0 million Ja-En training sentences, but because the data was automatically aligned, we use only the highest-confidence 2.0 million sentences to maintain the quality of the training data.

[5] http://github.com/neubig/wat2014

[6] http://code.google.com/p/egret-parser

[7] It may be possible to further improve translation accuracy by re-parsing the training data, but this comes at a significant computational cost, so in this work we only experiment with re-parsing the development and test corpora.

[8] While a large corpus is available, training the parser using the entire corpus is computationally expensive, so we randomly subsample a training corpus.

Table 2: Experiment results of Japanese-English translation

			Ja-En			Ja-Zh		
	Sentence selection	Tree selection	Sent	BLEU	RIBES	Sent	BLEU	RIBES
(a)	—	—	—	23.83	72.27	—	29.60	81.32
(b)	Random	Parser 1-best	96k	23.66	71.77	129k	29.75	‡ 81.55
(c)	Random	MT 1-best	97k	23.81	72.04	130k	29.76	‡ 81.53
(d)	Random	BLEU+1 1-best	97k	23.93	72.09	130k	‡ 29.89	‡ 81.66
(e)	BLEU+1 $\geq$ 0.7	BLEU+1 1-best	206k	‡ 24.27	72.38	240k	‡ 29.86	‡ 81.60
(f)	BLEU+1 $\geq$ 0.8	BLEU+1 1-best	120k	‡ 24.26	72.38	150k	‡ 29.91	81.47
(g)	BLEU+1 $\geq$ 0.9	BLEU+1 1-best	58k	‡ 24.26	72.49	82k	† 29.86	‡ 81.60
(h)	BLEU+1 Gain	BLEU+1 1-best	100k	† 24.22	72.32	100k	† 29.85	‡ 81.59
(i)	BLEU+1 $\geq$ 0.8 (Ja-En)	BLEU+1 1-best	—	—	—	120k	† 29.87	† 81.58

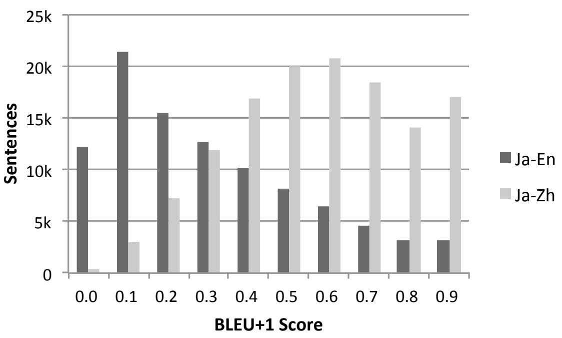


Figure 2: BLEU+1 score distribution of translations in Table 2 (d).

select the sentences randomly from the corpus as in Parser 1-best.

Oracle (BLEU+1≥t)
As described in Section 4.2.1, among the oracle translations and parse trees, we only use the sentences for which the BLEU+1 score exceeds the threshold t.

BLEU+1 Gain
As described in Section 4.2.2, among the oracle translations and parse trees, we use only the sentences which have a large difference of BLEU+1 score between 1-best and oracle translations. In this method, we maintain the sentence length distribution by using Formula (12) and (13).

It should be noted that when selecting the sentences randomly, we select $1/20$ of all training data in Japanese-English, $1/10$ in Japanese-Chinese translation. In BLEU+1 Gain, we select the top 100k sentences, which is a similar number of sentences as used in the other methods.

5.2. Experiment results

Table 2 shows the experimental results for Japanese-English and Japanese-Chinese translation. The dagger symbol in the table indicates that the translation accuracy of the proposed method is significantly higher than the baseline ($\dagger : p < 0.05$, $\ddagger : p < 0.01$). In Table 2 (b), (c), (d), the sentences for self-training are the same except where Egret fails to parse.[9] In Table 2, "Sent" indicates the number of sentences added through self-training and does not include the existing JDC data. In our analysis, we mainly focus on the BLEU score results, because we used BLEU+1 as a criterion when picking sentences for self training. Based on these results, we answer the following research questions:

- Is targeted self training through parse tree selection (Section 4.1) effective in improving translation results?
- Can sentence selection (Section 4.2) further reduce noise and improve accuracy?
- Is self-training language dependent, or portable across target languages?

Effect of Tree Selection: First, we can see that the method of using parser 1-best trees as self-training data did not achieve a BLEU score improvement in Ja-En and Ja-Zh translation (Table 2 (b)). Additionally, while in the MT 1-best method, the accuracy is improved compared to Parser 1-best in the Ja-En experiment, there is no improvement compared to the baseline system. In the Ja-Zh experiment, MT 1-best is almost the same as parser 1-best (Table 2 (c)). We manually analyzed the parse trees that have been used for self-training in these methods, and while there are some correct trees there are also many incorrect trees, which likely disturbed the training.

Next looking at the BLEU+1 1-best scores (Table 2 (d)), we can see that by selecting parse trees that were used in the oracle translations, BLEU scores slightly improved in both Ja-En and Ja-Zh experiments, with the Ja-Zh system significantly outperforming the baseline. Figure 2 shows the BLEU+1 score distribution of oracle translations used in self-training in this case. The label on the horizontal axis repre-

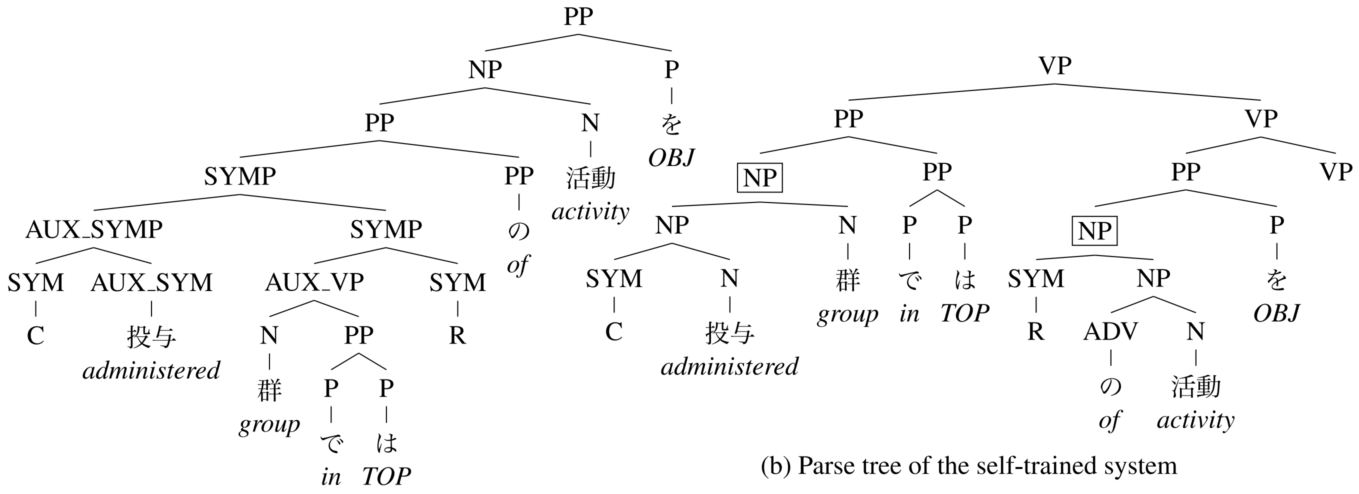
[9] In the methods except (b), we use Ckylark [23] trained by JDC as an alternative parser when Egret fails to parse.

Table 3: Self-trained Japanese parser accuracy

	Sentence selection	Tree selection	Recall	Precision	F-Measure
(a)	—	—	84.88	84.77	84.83
(b)	Random	Parser 1-best	86.52	86.41	† 86.46
(c)	BLEU+1 ≥ 0.8	BLEU+1 1-best	88.13	88.01	‡ 88.07

Table 4: An example of an improvement in Japanese-English translation

Source	C 投与 群 で は R の 活動 を ２４０ 分 に わたって 明らか に 増強 した 。
Reference	in the C - administered group , thermal reaction clearly increased the activity of R for 240 minutes .
Baseline	for 240 minutes clearly enhanced the activity of C administration group R .
BLEU+1≥0.8	for 240 minutes clearly enhanced the activity of R in the C - administration group .



(a) Parse trees of the baseline system

(b) Parse tree of the self-trained system

Figure 3: An example of an improvement in parsing result

sents the number of sentences which have BLEU+1 scores greater than x and less than $x + 0.1$, where x is the label. As can be seen from the figure, there are many sentences where even the oracle has a low score, motivating the sentence selection methods presented in Section 4.2.

Effect of Sentence Selection: Next, we examine the effect of selecting sentences using the BLEU+1 score threshold. From the results, we find it effective, with translation accuracy improving especially in Ja-En experiments (Table 2 (e),(f),(g)). In the Ja-Zh experiments, the BLEU+1 score distribution of oracle translations tends to be higher than in the Ja-En translations, explaining why this method is more effective in Ja-En experiments. From this result, we can say that when doing parser self-training, it is important to remove the low accuracy parse trees and keep only high accuracy parse trees from the training data. This is particularly true when there are a large number of oracle translations with low accuracies.

Moreover, by using the data that have a large BLEU+1 score improvement between MT 1-best and oracle translations, we achieved an effect similar to that of the BLEU+1 threshold method (Table 2 (h)).

Target Language Portability: Finally, we examine what happens when a parser used for translation in one language pair, Japanese-English, is used to parse the sentences for translation in another language pair, Japanese-Chinese (Table 2 (i)). Interestingly, the improvement in this case is quite similar to the parser trained directly on the Japanese-Chinese data. Thus, the model's dependence on target language is not strong, and it may be possible to do more effective self-training by using several target languages as training data.

5.3. Example of improved translation

Table 4 shows an example of an improvement caused by self-training in Japanese-English translation. In addition, Figure

3 shows the parse trees used in the translations in Table 4. In this sentence "C 投与 群" (C administration group) and "R の 活動" (activity of R) are noun phrases. The baseline parser cannot identify noun phrases correctly, and translation is affected by this parse error. On the other hand, the self-trained parser can identify noun phrases correctly, resulting in these phrases being correctly translated.

5.4. Self-trained parser accuracy

We also performed experiments to examine the parser accuracy itself. We manually created 100 reference parse trees from the Ja-En ASPEC test data, and checked the accuracy of the baseline and self-trained parsers with respect to these trees by using Evalb.[10] Table 3 shows the experimental results. The dagger symbol in the table indicates that the F-Measure of the proposed method is significantly higher than the baseline ($\dagger : p < 0.05,\ \ddagger : p < 0.01$).

Here, we can see that the parser 1-best method achieved significantly higher accuracy than the baseline at the 95% level. In addition, our proposed targeted self-trained parser could achieve a further significant gain in accuracy. These results show that our proposed targeted self-training methods improve not only MT results, but also parser accuracy itself.

6. Conclusion

In this study, we proposed a targeted self-training method for syntactic parsers used in syntax-based MT, and verified its effect on T2S translation. We performed experiments on Japanese-English and Japanese-Chinese translation and found that by using the self-trained parser that we were able to achieve a significant improvement in the accuracy of a state-of-the-art translation system. Moreover, we found that the model self-trained by Japanese-English sentences can also contribute to more accurate Japanese-Chinese translations.

Our future work includes verifying that this method can be used for other languages pairs. Moreover, the experimental results suggest that the effect of self-training does not heavily depend on the source language, and thus it may be possible to improve the translation accuracy by applying self-training over data from multiple languages pairs. Furthermore, we will test the effect on translation accuracy when performing multiple iterations of parser self-training, or using the self-trained parser to re-parse the training data.

7. Acknowledgments

Part of this research was supported by JSPS KAKENHI Grant Number 25730136 and 24240032.

[10] http://nlp.cs.nyu.edu/evalb

8. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. HLT*, 2003, pp. 48–54.

[2] K. Yamada and K. Knight, "A syntax-based statistical translation model," in *Proc. ACL*, 2001, pp. 523–530.

[3] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proc. ACL*, 2006, pp. 609–616.

[4] G. Neubig and K. Duh, "On the elements of an accurate tree-to-string machine translation system," in *Proc. ACL*, 2014, pp. 143–149.

[5] H. Mi, L. Huang, and Q. Liu, "Forest-based translation," in *Proc. ACL*, 2008, pp. 192–199.

[6] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *Proc. HLT*, 2006, pp. 152–159.

[7] J. Katz-Brown, S. Petrov, R. McDonald, F. Och, D. Talbot, H. Ichikawa, M. Seno, and H. Kazawa, "Training a parser for machine translation reordering," in *Proc. EMNLP*, 2011, pp. 183–192.

[8] J. Graehl and K. Knight, "Training tree transducers," in *Proc. HLT*, 2004, pp. 105–112.

[9] H. Zhang and D. Chiang, "An exploration of forest-to-string translation: Does translation help or hurt parsing?" in *Proc. ACL*, 2012, pp. 317–321.

[10] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[11] E. Charniak, "Statistical parsing with a context-free grammar and word statistics," in *Proc. AAAI*, 1997, pp. 598–603.

[12] Z. Huang and M. Harper, "Self-training PCFG grammars with latent annotations across languages," in *Proc. EMNLP*, 2009, pp. 832–841.

[13] D. Talbot, H. Kazawa, H. Ichikawa, J. Katz-Brown, M. Seno, and F. Och, "A lightweight evaluation framework for machine translation reordering," in *Proc. WMT*, 2011, pp. 12–21.

[14] G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, and F. Casacuberta, "Does more data always yield better translations?" in *Proc. ACL*, 2012, pp. 152–161.

[15] G. Neubig, "Forest-to-string SMT for asian language translation: NAIST at WAT2014," in *Proc. WAT*, 2014.

[16] T. Nakazawa, H. Mino, I. Goto, S. Kurohashi, and E. Sumita, “Overview of the 1st Workshop on Asian Translation,” in *Proc. WAT*, 2014.

[17] G. Neubig, “Travatar: A forest-to-string machine translation engine based on tree transducers,” in *Proc. ACL Demo Track*, 2013, pp. 91–96.

[18] S. Mori, H. Ogura, and T. Sasada, “A Japanese word dependency corpus,” in *Proc. LREC*, 2014, pp. 753–758.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proc. ACL*, 2002, pp. 311–318.

[20] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, “Automatic evaluation of translation quality for distant language pairs,” in *Proc. EMNLP*, 2010, pp. 944–952.

[21] C.-Y. Lin and F. J. Och, “Orange: a method for evaluating automatic evaluation metrics for machine translation,” in *Proc. COLING*, 2004, pp. 501–507.

[22] P. Koehn, “Statistical significance tests for machine translation evaluation,” in *Proc. EMNLP*, 2004, pp. 388–395.

[23] Y. Oda, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, “Ckylark: A more robust PCFG-LA parser,” in *Proc. NAACL*, 2015, pp. 41–45.