Analysis of Back-translation Methods for Low-Resource Neural Machine Translation

Nuo Xu[1], Yinqiao Li[1], Chen Xu[1], Yanyang Li[1], Bei Li[1],

Tong Xiao[1,2] and Jingbo Zhu[1,2]

[1] Northeastern University, Shenyang, China
[2] NiuTrans Research, Shenyang, China
{xunuo0629,xuchen}@stumail.neu.edu.cn,
li.yin.qiao.2012@hotmail.com, {blamedrlee,libei_neu}
@outlook.com, {xiaotong,zhujingbo}@mail.neu.edu.cn

Abstract. Back translation refers to the method of using machine translation to automatically translate target language monolingual data into source language data, which is a commonly used data augmentation method in machine translation tasks. Previous researchers' works on back translation only focus on rich resource languages, while ignoring the low resource language with different quality. In this paper, we compare various monolingual selection methods, different model performance, pseudo-data and parallel corpus ratios, and different data generation methods for the validity of pseudo-data in machine translation tasks. Experiments on Lithuanian and Gujarati, two low-resource languages have shown that increasing the distribution of low-frequency words and increasing data diversity are more effective for models with sufficient training, while the results of insufficient models are opposite. In this paper, different back-translation strategies are used for different languages, and compared with common back-translation methods in WMT news tasks of two languages, and the effectiveness of the strategies is verified by experiments. At the same time, we find that combined back-translation strategies are more effective than simply increasing the amount of pseudo-data.

Keywords: low-resource , back-translation , machine translation.

1 Introduction

Neural machine translation model based on large volume dataset [1] to learn the mapping between the source and the target has advanced the state-of-the-art on various language pairs [2]. The availability of massively parallel corpus is critical to train strong systems. But the resource of massive bitext is limited and there are abundant of monolingual sentences which can be furthermore leveraged to increase the amount of bilingual corpus [3]. Back-translation is a common and effective data augmentation method without modification to the training strategy, which we translate the target monolingual data into the source by reverse translation models. Surprisingly, researchers find the pseudo-data ratio and the improvement of translation quality is non-linear related [4]. There are several research on selecting the appreciate monolingual data from the mass data, such as extracting the monolingual data according to the word frequency or the words with high loss [4], or scoring by language models to pick in-domain corpus [5]. For the data generation method, it is not limited to the beam search manner, but based on sampling method [6] or noisy beam search to generate the synthetic data.

In the translation tasks of low resource language pair, the quantity and quality of parallel corpora are more unsatisfactory than the rich-source language, which makes the translation model obtained by using bilingual data less robust and accurate. For example, some methods that can effectively improve the translation performance in rich resource translation tasks may not achieve the desired improvement on condition of low resource languages. Therefore, the research on the back-translation strategy for low-resource language has not been widely investigated. In this paper we verify the effectiveness of various back-translation strategies on low-resource translation tasks through experimenting and analyzing.

We focus on BT strategies from four aspects as the following: how to select the monolingual data, the methods to generate pseudo corpus, the translation performance of target-to-source translation model, and the ratio of synthetic data in the bilingual corpus. We investigate the data selection method according to the frequency of rare words and high predictive loss words, or scoring by in-domain language model. For the performance of pseudo-data generation models, we explore the effect of different convergence states and hyper-parameter settings. Additionally, we furthermore analyze several synthetic source sentence generation methods, including random or restricted sampling from the model distribution or conventional beam search. We experimented different strategies on two low-resource language tasks, including Lithuania to English and Gujarat to English. Our analysis shows that on condition of 1:1 between the synthetic data and real bitext, we achieved 1.5/0.5 BLEU improvements on the validation set and the test set respectively. Similarly, 0.4/0.7 BLEU improvements were obtained on Gujarat to English language pair. Meanwhile, we found that the appropriate back translation strategy improves the translation performance effectively than simply increasing the synthetic data volume.

2 Related Work

2.1 Back translation

Neural machine translation model solely based on parallel corpus has achieved promising performance. Due to the limitation of bitext resource, researchers often prefer leveraging the abundant monolingual data to furthermore enhance the translation quality. BT is one of the data augmentation methods which is simple and effective and wildly used among the researchers. Experiment results show that even if the parallel corpus is rich enough, using the back-translation method can also effectively improve the translation quality.

2.2 Data Generation Strategy

Beam search Beam search retains several candidates in each decoding step, and finally obtains the approximate global optimal translation. This method is more accurate than the ordinary greedy search algorithm, which only retains the word with the highest probability in each decoding step, but for the entire target sentence, it may be not the global optimal choice. The beam search alleviates this problem by collecting several high-probability words into the consideration to participate in the next prediction, so as to obtain the target hypothesis with the highest overall probability [10]. Equation 1 represents the decoding algorithm where is the length of the target sentence and is the length of the source sentence:

$$y = \underset{y}{\operatorname{argmax}} P(y^{<1>}, y^{<2>}, y^{<3>}, \ldots y^{<T_y>} | x^{<1>}, x^{<2>}, \ldots x^{<T_x>}) \tag{1}$$

TopK probability search The TopK-based decoding method is a translation-based probability decoding method. The decoding process is divided into two steps. Firstly, the method selects words with the highest probability according to the distribution of the target vocabulary. The second step is to sample the words restricted from the highest-N candidates among the vocabulary [6]. This method slightly increases the diversity of the prediction distribution compared with beam search. However, the pseudo data generated by this method is not much different from beam search due to the constrained sampling space.

Sampling probability search Sampling is a more flexible probabilistic decoding method. This method removes the restriction on the candidate set in the TopK search mode, and randomly selects words from the whole vocabulary distribution [6]. In this way, we can gain more diverse and noisy synthetic data than beam search or restricted sampling, which can improve the robustness of the model by the source side inaccurate samples. Nevertheless, lots of noise brought into the model may damage the model precision.

Compared with the beam search, TopK and Sampling as Fig.1：

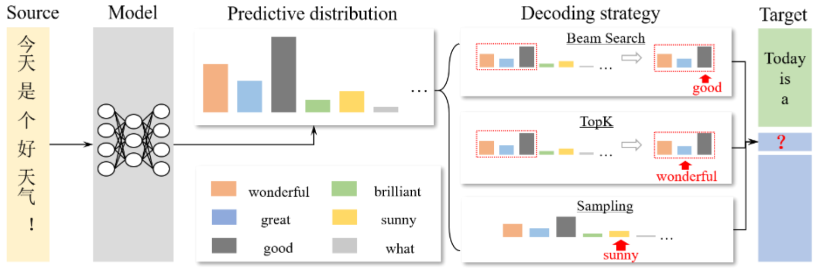

Fig. 1. Contrast diagram of beam search, TopK and Sampling

3 Experiments

3.1 Data Processing

This paper experiments various back-translation strategies in the Lithuanian-English (Lt-En), Gujarat-English (Gu-En) news translation tasks from the 2019 WMT (Workshop on Machine Translation). The parallel corpus in the Lt-En task comes from Europarl v9, ParaCrawl v3, and Rapid corpus of EU press releases. In the Gu-En task, it comes from Bible Corpus, crawled corpus, Localisation extracted from OPUS and parallel corpus extracted from Wikipedia. The English monolingual data is from News Crawl 2015-2018. We filter the parallel corpus and monolingual data by garbled filtering, length ratio filtering, word alignment, language model scoring [9], deduplication and etc.

After the above filtering steps, we retain nearly 1.9M parallel corpus in the Lt-En task, 79K parallel corpus in the Gu-En task and 10M English monolingual data. We use the sacrebleu.perl [10] script to calculate the BLEU as the evaluation metric of translation quality. We perform BPE on the training data [11], and the number of merge operations is 32k.

There are only a few test sets available for low resource language pairs, i.e., the validation set and test sets issued by WMT19. According to our preliminary analysis, the small number of test set sentences, the huge distinction between the construction of the validation set and training data together with the diverse domains involved in the content, will have a great impact on the evaluation of the models. So we randomly cut out 2000 bilingual sentence pairs from the parallel corpus as our test set, which shares the same distribution as the training data, for reverse translation model experiments (4.2) and data generation method experiments (4.3).

For simplicity, Train denotes the test set extracted from the training data, Dev denotes the official validation set, and Test denotes the official test set.

3.2 Baselines

This paper employs the Transformer [12] model for the experiments. We use Tensor2Tensor and Fairseq open source system for model training and decoding. There are 6 layers in both encoder and decoder. The hidden layer dimension is 512 and 1024 and the attention number is 8 and 16 for the Transformer_base and the Transformer_big setting respectively. The batch size is 4096, and the maximum sentence length is 250. The residual dropout is 0.1, the initial learning rate is set to 0.001, and the optimizer is Adam. We train 15 epochs for Transformer_base, 30 epochs for Transformer_big, and average the last 5 checkpoints. During the experiment, 8 GPU devices is used for model training. The length penalty used to generate Lithuanian pseudo data is 0.7 and 1.0 for Gujarat. The Transformer_base baseline scores are shown in Table 1:

Table 1. Bidirectional baseline performance

Language direction	Corpus size	BLEU	
		Dev	Test
Lt-En	1.9M	27.1	29.2
Gu-En	79K	3.2	3.5

4 Analysis

4.1 Pseudo Dataset Size

This part of the experiment is based on the Transformer_base setting. To investigate the impact of the pseudo data size on the scarce resources scenario, we experiment with the Gujarati language, as it has much less parallel data. By varying the ratio of parallel corpus to pseudo data, we observe the influence of different pseudo corpus scales on the model performances. The experimental results are shown in Fig. 2:

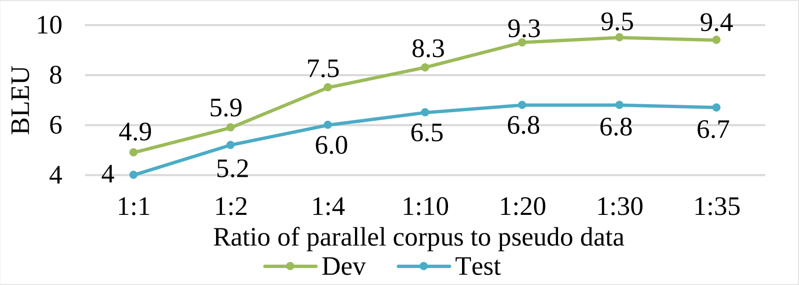

Fig. 2. The effect of pseudo data ratio on model performance

According to the experimental results, we find that the model performance does not grow linearly with the pseudo dataset size. The positive effect of the pseudo dataset size becomes marginalized when more pseudo data are added. We believe that this is due to the quality of the pseudo data. The pseudo data is generated by the translation model trained on the parallel corpus, so its quality is impossible to beat the parallel corpus. Adding excessive pseudo data reduces their effectiveness.

4.2 Reverse Translation Models

This part of the experiment makes use of the larger Lt-En corpus. It compares how pseudo data generated from the reverse translation models with different convergence states affect the performance. The performances of the reverse translation models are shown in Table 2:

The pseudo data generated by these four models are mixed with the parallel corpus in turn. The experimental results are shown in Table 3. During the convergence, the neural network fits the training data progressively. If the pseudo data from the insufficiently trained model are directly used, then the mismatch between the data distribution and the pseudo data distribution will have a negative effect on the final performance. In addition, we also find from Table 2 and Table 3 that the convergence state of the model affects the accuracy of the model prediction more even the performance gap among the converged models of different scales is similarly remarkable.

Table 2. Reverse translation model performance

Parameter	Epoch	Convergence state	BLEU	
			Dev	Test
Transformer_base	3epochs	Not converged	15.1	9.5
Transformer_base	8epochs	Not fully converged	19.9	11.3
Transformer_base	15epochs	Fully converged	20.2	11.5
Transformer_big	30epochs	Fully converged	21.0	12.7

For different scaled reverse translation models, the better the performance, the more the performance boost from the corresponded pseudo dataset are observed. However, high reverse translation model performance does not result in significant performance distinction on the models trained with the corresponding pseudo data.

Table 3. The effect of different model performance on pseudo data

Pseudo data generation model	Corpus size	BLEU		
		Dev	Test	Train
—	1.9M	27.1	29.2	45.1
Transformer_base in unconverged state	3.9M	28.6	29.2	45.4
Transformer_base in not fully converged	3.9M	29.5	29.4	46.2
Transformer_base in fully converged	3.9M	30.2	29.8	46.4
Transformer_big in fully converged	3.9M	30.6	29.1	46.3

4.3 Pseudo Data Generation Methods

This part of the experiment is based on the Transformer_base model. English monolingual data are randomly selected and the generation methods are beam search, TopK, and Sampling. The experimental results are shown in Table 4:

We can see that the Sampling method obtains the best performance over the others in Lt-En tasks. It is possibly due to the fact that Sampling introduces noisy data while in one way enriches the linguistic phenomenon within the training data, making the model more robust. For a translation task with sufficient parallel sentence pairs, the considerable amount of parallel data prevents the model from performance degradation led by the sampling noise. In contrast, Sampling method became the worst one in the Gu-En task, which has much less training data than Lt-En. We hypothesize that it is the result of which the reverse translation model trained on the small parallel corpus is unable to generate high quality translation for back-translation. Moreover, insufficient parallel data could be easily overwhelmed by the huge amount of noise within the pseudo corpus and thus cannot serve as a good training signal. Model will therefore be misled by the noise and degrade severely.

Table 4. The effect of different data generation methods on pseudo data

Language direction	Generation method	Corpus size	BLEU		
			Dev	Test	Train
Lt-En	—	1.9M	27.1	29.2	45.1
	Beam search	3.9M	30.2	29.8	46.4
	TopK	3.9M	30.3	29.5	45.9
	Sampling	3.9M	31.6	30.1	45.5
Gu-En	—	79K	3.2	3.5	27.1
	Beam search	160K	4.8	4.0	22.3
	TopK	160K	4.6	3.9	22.1
	Sampling	160K	3.4	3.1	20.6

On the Train test set, since the Sampling method diversifies the pseudo corpus hence be inconsistent with the training data, its results were the worst on the Train test set. This gap is even large in Gu-En task, where the monolingual data are out of the parallel data domain.

To investigate the reason why the Sampling method is more effective in the Lt-En task than the Beam Search, we analyze their pseudo data as well as the parallel corpus. We first construct the vocabulary from the parallel corpus and collect word statistics of the parallel corpus, the pseudo corpus generated by both Beam Search and Sampling. We then group every 2K words as a whole and present the frequency results of the last seven groups in Fig. 3：

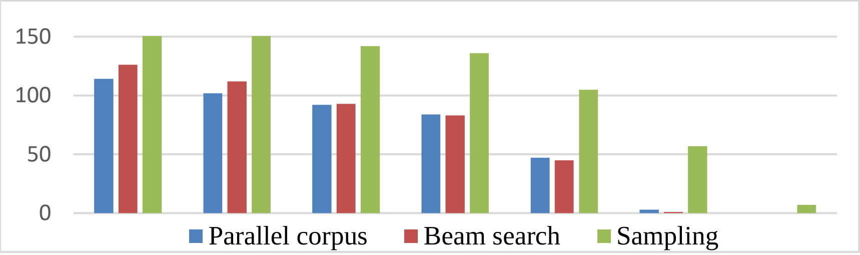

Fig. 3. Data distribution

As can be discovered in Fig. 3, the word frequency distribution of the Beam Search is very similar to the one of the parallel corpus, and the pseudo data generated by Sampling contains more low frequency words. This is because the Sampling method randomly selects words during generation, which is equivalent to assigning non-zero probability to low-frequency words compared to all zero in Beam Search, making the data distribution of pseudo data more diverse. Therefore, we suspect that one of the key reasons for the effectiveness of the Sampling method is encouraging the occurrence of low frequency words in the source language. We conduct several experiments to verify our hypothesis, and the results of which are shown in Table 5:

Table 5. Results of the comparison experiment related to the Sampling method

Language direction	Generation method	Corpus size	BLEU	
			Dev	Test
Lt-En	Beam search	3.9M	30.2	29.8
	Sampling	3.9M	31.6	30.1
	Beam search + Sampling	3.9M	31.4	29.8
	Low frequency words + Beam search	3.9M	30.8	29.6

We extract the sentence pairs (1.1M) from the pseudo data generated by Sampling method, which contains the source-side low-frequency words in the parallel corpus, and deduplicate it with the previous extracted 2M English monolingual data. Then we extract sentence pairs that share the same English part from the pseudo data generated by Beam Search. It can be seen from the Table 5 that mixing these two kinds of pseudo data has 1.2 BLEU performance improvement over only Beam Search method, and is only inferior to the result of Sampling method about 0.2 BLEU. We further select samples with Lithuanian low-frequency words aligned and find the result of the Dev set is improved by 0.6 BLEU. Therefore, we believe that the effectiveness of Sampling method comes in two ways, where it increases the occurrences of low-frequency words for the source-side and the diversity of pseudo data through randomness. However, we believe that the pseudo data generated by Sampling requires good performance of the baseline model to against the noise introduced in Sampling pseudo data.

4.4 Monolingual Data Selection Strategies

This section describes the impact of various monolingual selection strategies on the back-translation method. Previous studies have shown that in the training process, words that are difficult to be predicted on the target language side tend to be more accurate after adding pseudo data [4]. Therefore, we select sentences containing these difficult words as the monolingual data for generating the pseudo corpus. These difficult words include the low-frequency words of both the source and target languages and the target language words with high prediction loss. For the source-side low-frequency words, we use the GIZA++ [13] word alignment tool to find their corresponding target-side words in order to select the target-side monolingual sentences. Based on the above definitions of difficult words, three monolingual corpuses containing these three types of difficult words are selected for experiments. At the same time, we also use the language model to select the monolingual data. The language model architecture is Transformer_base, and the data generation method is beam Search. The experimental results are shown in Table 6:

According to Table 6, it is found that in the Lt-En task, the target-side low-frequency words and the ones with high target-side prediction loss are not as good as the random selection baseline. Compared with the random selection baseline, using the data selected by the source-side low-frequency words and the language model improve 0.6 and 0.7 BLEU on the Dev set, and 0.4 and 0.6 BLEU on the Test set respectively. In the Gu-En task, all four methods outperform the random selection baseline. The best experimental results are achieved by using the source-side low-frequency words and the language model. We believe that target-side data quality plays a crucial role in model training as it ensures the target correctness. The higher the quality, the better the model fits then the better the performance. Increasing low-frequency words in the source-side forces the model to fit them better and thus more robust to their appearance. However, the use of the target-side low-frequency words for monolingual data selection may result in a large number of low-frequency words in the generated translations and potentially degrade the performance.

Table 6. The effect of different monolingual selection methods on pseudo data

Selection method	Language direction	BLEU		Language direction	BLEU	
		Dev	Test		Dev	Test
Random	Lt-En	30.2	29.8	Gu-En	4.8	4.0
Freq(target)		30.0	29.8		5.0	4.3
Freq(source)		30.8	29.6		5.2	4.6
Loss		29.9	28.6		5.3	4.7
Language model		30.9	29.8		5.2	4.8

4.5 General Results

Based on our previous observations, this section combines the most effective pseudo data generation methods with the monolingual selection strategies. Under the Transformer_base model setting, the experimental results are shown in Table 7. We find that selecting only 880K samples that contain the high prediction loss word have even more significant improvement than the 1.6M counterpart on the Test set.

Table 7. Combined experimental results

Experimental description	Language direction	Coupus size	BLEU	
			Dev	Test
Language model+Sampling	Lt-En	3.9M	31.7	30.3
Freq(source)+Sampling	Lt-En	3.9M	31.5	30.3
Language model + Beam search	Gu-En	160K	5.2	4.8
Loss+ Beam search	Gu-En	160K	5.3	4.7
Synthetic(1:10) + Loss + Beam search	Gu-En	880K	9.1	6.9

5 Conclusion

We investigate the effect of several strategies on back-translation, including synthetic-data ratio, reverse translation model performance and different synthetic generation methods. Experiment results show that enlarge the synthetic volume can significant improve the translation quality on low resource languages. We find convergence state impacts more on the quality of synthetic data than the performance of target-to-source translation model. For the generation method, sampling is more helpful when the translation model is strong enough, while beam search shows more benefits when the model converges insufficiently.

6 Acknowledgments

This work was supported in part by the National Science Foundation of China (Nos. 61876035, 61732005 and 61432013) and the National Key R&D Program of China (No. 2019QY1801).

References

1. Bahdanau D, Cho K, Bengio Y, et al. Neural Machine Translation by Jointly Learning to Align and Translate[J]. international conference on learning representations, 2015.
2. Luong T, Pham H, Manning C D, et al. Effective Approaches to Attention-based Neural Machine Translation[J]. empirical methods in natural language processing, 2015: 1412-1421.
3. Sennrich R, Haddow B, Birch A, et al. Improving Neural Machine Translation Models with Monolingual Data[J]. meeting of the association for computational linguistics, 2016: 86-96.
4. Fadaee M, Monz C. Back-Translation Sampling by Targeting Difficult Words in Neural Machine Translation[J]. empirical methods in natural language processing, 2018: 436-446.
5. Amittai Axelrod, Yogarshi Vyas, Marianna Martindale, Marine Carpuat, and Johns Hopkins. Classbased n-gram language difference models for data selection.2015. In IWSLT (International Workshop on Spoken Language Translation), pages 180–187.
6. Edunov S, Ott M, Auli M, et al. Understanding Back-Translation at Scale[J]. empirical methods in natural language processing, 2018: 489-500.
7. Lopez A. Statistical machine translation[J]. ACM Computing Surveys, 2008, 40(3).
8. Graves A. Sequence Transduction with Recurrent Neural Networks[J]. arXiv: Neural and Evolutionary Computing, 2012.
9. Moore R C, Lewis W D. Intelligent Selection of Language Model Training Data[C]. meeting of the association for computational linguistics, 2010: 220-224.
10. Post M. A Call for Clarity in Reporting BLEU Scores.[J]. arXiv: Computation and Language, 2018: 186-191.
11. Sennrich R, Haddow B, Birch A, et al. Neural Machine Translation of Rare Words with Subword Units[J]. meeting of the association for computational linguistics, 2016: 1715-1725.
12. Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[J]. neural information processing systems, 2017: 5998-6008.
13. Brown P F, Pietra V J, Pietra S D, et al. The mathematics of statistical machine translation: parameter estimation[J]. Computational Linguistics, 1993, 19(2): 263-311.