information

MDPI

Article

A Diverse Data Augmentation Strategy for Low-Resource Neural Machine Translation

Yu Li [1,2,3], Xiao Li [1,2,3,*], Yating Yang [1,2,3] and Rui Dong [1,2,3]

1 Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China; liyu315@mails.ucas.edu.cn (Y.L.); yangyt@ms.xjb.ac.cn (Y.Y.); dongrui@ms.xjb.ac.cn (R.D.)
2 University of Chinese Academy of Sciences, Beijing 100049, China
3 Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China
* Correspondence: xiaoli@ms.xjb.ac.cn; Tel.: +86-136-599-13514

Received: 9 March 2020; Accepted: 4 May 2020; Published: 6 May 2020

Abstract: One important issue that affects the performance of neural machine translation is the scale of available parallel data. For low-resource languages, the amount of parallel data is not sufficient, which results in poor translation quality. In this paper, we propose a diversity data augmentation method that does not use extra monolingual data. We expand the training data by generating diversity pseudo parallel data on the source and target sides. To generate diversity data, the restricted sampling strategy is employed at the decoding steps. Finally, we filter and merge origin data and synthetic parallel corpus to train the final model. In the experiment, the proposed approach achieved 1.96 BLEU points in the IWSLT2014 German–English translation tasks, which was used to simulate a low-resource language. Our approach also consistently and substantially obtained 1.0 to 2.0 BLEU improvement in three other low-resource translation tasks, including English–Turkish, Nepali–English, and Sinhala–English translation tasks.

Keywords: neural machine translation; back translation; data argument; low resource language

1. Introduction

Neural machine translation (NMT) is one of the most interesting areas in natural language processing (NLP). It is based on an encoder–decoder architecture, where the encoder encodes the source sentence as a continuous space representation, and the decoder generates the target sentence based on the encoder output [1]. NMT has achieved tremendous success in the past few years [2,3]. It usually requires a large amount of high-quality bilingual corpus for training [4]. For some languages, there are not enough resources to train a robust neural machine translation system. Therefore, studies in neural machine translation under low-resource conditions is an enormous challenge. There have been many studies on low-resource language machine translation. Using high-resource language to help improve the performance of low-resource neural machine translation is an intuitive approach; for example, transfer learning [5], model-agnostic meta-learning algorithm [6], triangular architecture [7], and multi-way, multilingual NMT frameworks [8]. As an essential way to enhance translation performance by generating additional training samples, data augmentation has been proven useful for low-resource neural machine translation. Some previous works use a synonym to replace specific words in training data. However, thesauruses are scarce for low-resource languages. Another approach is based on replacing some words in the target sentence with other words from the target vocabulary, for instance, randomly replacing the word with a placeholder or sampling the word from the frequency distribution of the vocabulary [9] and randomly setting the word embedding to 0 [10]. However, due to data sparsity in low-resource languages, it is difficult for these methods to leverage all possible augmented data. Another data augmentation method is the use of monolingual data. The well-known methods are back-translation

and self-learning [11–13]. Back-translation is a data augmentation approach that translates monolingual data of the target side into the source to augment pseudo bitext [11]. Zhang et al. [12] proposed a self-learning method. They proved that translating the monolingual source data into the target in order to augment the training data is useful for improving the translation performance. However, these methods require substantial efforts to collect and clean the necessary amount of monolingual data. Meanwhile, the additional monolingual corpus is scarce in some low-resource languages.

In this paper, we propose an effective data augmentation strategy that does not use any monolingual data, which augments the training data by generating diversity source and target sentences on the origin data. Compared with the original training data, the diversity source or target data has the same semantics but different expressions [14,15]. To augment diversity data, we train the translation model in two directions: backward (target-to-source) and forward (source-to-target). Then, these translation models are employed to decode the training data multiple times. In the decoding process, we use the restricted sampling strategy, which can produce the diversity data. Finally, the duplicate sentences are deleted in the training data and pseudo parallel data to train the final model.

To demonstrate the effectiveness of our method, we first performed experiments on the IWSLT2014 German–English translation tasks, which can be used to simulate a low-resource setting. We compare our approach with other data augmentation methods, the results on English–German translation tasks show that the proposed data augmentation method achieved an improvement of 1.96 BLEU points over the baselines without using extra monolingual data. Our method achieved the best results among all data augmentation methods. We also conducted experiments on three low-resource translation tasks, including English–Turkish, Nepali–English, and Sinhala–English. The experimental results indicate that our method also boosted performance by 1.51, 1.28, and 1.53 BLEU points in the English–Turkish, Nepali–English, and Sinhala–English translation tasks, respectively.

In a summary, our contributions are as follows: (1) We propose a data augmentation strategy that has proved effective for many languages. (2) Compared with other data augmentation approaches, ours obtained the best result. (3) We performed experiments to explain the effectiveness of our method. These results verify that the increase in performance is not due to data replication, and our approach can produce diverse data. Finally, we found that the backward model was more important than the forward model.

In the rest of this article, Section 2 presents some related works about data augmentation. Section 3 describes the details of the restricted sampling strategy and our diversity data augmentation. The experiment details and results are shown in Section 4. Section 5 presents some experiments we conducted to analyze the effect of our data augmentation method. Finally, the conclusions are presented in Section 6.

2. Related Work

Although neural machine translation (NMT) has achieved better performance in many languages, data sparsity and the lack of morphological information are important issues. There are some works which aimed to improve the effectiveness of machine translation, including adjusting the translation granularity or incorporating morphological information. Sennrich et al. used the byte-pair-encoding algorithm to segment source sentences or target sentences into subword sequences [16]. Pan et al. segmented words into morphemes based on morphological information. Their results show that their method can effectively reduce data sparsity [17]. Sennrich et al. improved the performance of their encoder by employing some source side features, such as morphological features, part-of-speech tags, and syntactic dependency labels [18]. Tamchyna et al. employ an encoder–decoder to predict a sequence of interleaving morphological tags and lemmas, then use a morphological generator to generate the final results [19]. However, for some languages, there is a lack of effective morphological analysis tools. Therefore, some researchers pay more attention to improving the performance of machine translation with data augmentation.

Data augmentation is an effective method that generates additional training examples to improve the performance of deep learning. This method has been widely applied in many areas. In the field of computer vision, some image augmentation methods such as cropping, rotating, scaling, shifting, and adding noise are widely used and highly effective [20,21]. There are several related works about data augmentation for NMT. One of the data augmentation methods is based on word replacement. Fadaee et al. propose a word replacement method that uses the target language model to replace the high-frequency words with rare words, then changes its corresponding word in the source [22]. Xie et al. replace the word with a placeholder token or a word sampled from the frequency distribution of the vocabulary [9]. Kobayashi et al. use a wide range of substitute words generated by a bi-directional language model to replace the word token in the sentence [23]. Wu et al. replace the bi-directional model with BERT [24], which is a more powerful model, then use it to generate a set of substitute words [25]. Gao et al. propose a soft contextual data augmentation method that uses a soft distribution to replace the word representation instead of a word token [26]. Due to the data sparsity for low-resource languages, it is difficult for those methods to leverage all possible augmented data.

The other category of data augmentation is based on monolingual data. Sennrich et al. propose a simple and effective data augmentation method, where the target language data is translated into the source to augment the parallel corpus [11]. It has been proved effective by many works [27–30]. Zhang et al. propose a self-learning data augmentation method that translates monolingual source data to target, then combines it with origin data to train the final model [12]. Imamura et al. show that generating synthetic sentences based on sampling is more effective than beam search. They generate multiple source sentences for each target [14]. Currey et al. show that copying target monolingual data into the source can boost the performance of low-resource translation [31]. Chang et al. and He et al. employ monolingual corpora from source and target sides to extend the back-translation method as dual learning [32,33]. A similar method has been applied in unsupervised NMT [34,35]. Hoang et al. suggest an iterative data augmentation procedure that continuously improves the quality of the back-translation and final systems [36]. Niu et al. use multilingual NMT, which trains two directions of a translation model in a single model to translate monolingual data that comes from source or target to generate synthetic data [37]. Zhang et al. propose a corpus augmentation method; they segment long sentences based on word alignment and use back-translation to generate pseudo-parallel sentence pairs [38]. Although these methods have significantly improved the effectiveness of machine translation, they use additional monolingual corpora.

3. Approach

In this section, we describe our diversity data augmentation approach in detail. First, we introduce the main idea of back-translation and self-learning. Then, we present a decoder strategy which is used to generate diversity data. Finally, the training process of our data augmentation approach is presented in detail.

3.1. Back-Translation and Self-Learning

Back-translation is an effective way to improve the performance of machine translation. It is usually used to increase the size of parallel data. Given the parallel language pairs $D = \{(s_n, t_n)\}_{n=1}^{N}$ and a monolingual target dataset $D_{mon} = \{t_{mon}^{m}\}_{m=1}^{M}$, the main idea of back-translation is as follows. First, the reverse (target-to-source) translation model $NMT_{T\to S}$ is trained with parallel corpus D. Then, the reverse translation model $NMT_{T\to S}$ is used to translate the monolingual target data D_{mon} into source. The source-language translation is denoted as $D_{st} = \left\{s_{st}^{m}\right\}_{m=1}^{M}$. The monolingual target data D_{mon} and its translation D_{st} are paired as synthetic bitext $D_{synthetic} = \left\{\left(s_{st}^{m}, t_{mon}^{m}\right)\right\}_{m=1}^{M}$. Finally, the initial corpus is combined with the synthetic corpus to train the main translation system $NMT_{S\to T}$.

The main idea of self-learning is the same as back-translation. The difference is that self-learning is based on monolingual source data. Given the parallel language pairs $D = \{(s_n, t_n)\}_{n=1}^{N}$ and a

monolingual source dataset $D_{mon} = \{s^m_{mon}\}^M_{m=1}$, the process of self-learning includes the following steps: First, the forward translation model $NMT_{S\to T}$ is trained with parallel corpus D. Second, the monolingual source data D_{mon} are translated into the target by translation model $NMT_{T\to S}$. The monolingual source data and its translations are combined as synthetic corpus $D_{synthetic} = \left\{\left(s^m_{mon}, t^m_{st}\right)\right\}^M_{m=1}$. Finally, the main translation system $NMT_{S\to T}$ is trained with the mixture of parallel and synthetic data.

3.2. Decoder Strategy

NMT systems typically use beam search to translate the sentences [39]. Beam search is an algorithm that approximately maximizes conditional probability. Given the source sentence, it retains several high-probability words at each decoding step, and generates the translation with the highest overall probability:

$$y_t = \text{argmax}(\Pr(y_t|y_{<t}, x)) \tag{1}$$

However, beam search always focuses on the head of the model distribution, which results in very regular translation hypotheses that do not adequately cover the actual data distribution [30]. On the contrary, decoding based on sampling or restricted sampling can produce diverse data by sampling from the model distribution [14,30]. At each decoding step, the sampling method randomly chooses the token from the whole vocabulary distribution as:

$$y_t = sampling(\Pr(y_t|y_{<t}, x)), \tag{2}$$

where $sampling_y(P)$ denotes the sampling operation of y according to the probability of distribution P.

The restricted sampling strategy is a middle ground between beam search and unrestricted sampling. It adds a restriction on the selection of candidate tokens. The process of the restricted sampling strategy contains the following steps: First, at each decoding step, the translation model generates the probability distribution P. According to the output distribution P, it selects the k-highest probability tokens as candidate set C. Second, it renormalizes all words in the candidate set. Finally, it samples a token from the candidate set as output:

$$C = \underset{k}{\text{argmax}}\left(\Pr(y_t|y_{<t}, x)\right), \tag{3}$$

$$y_t = sampling(C). \tag{4}$$

Figure 1 shows the difference between these decoding strategies.

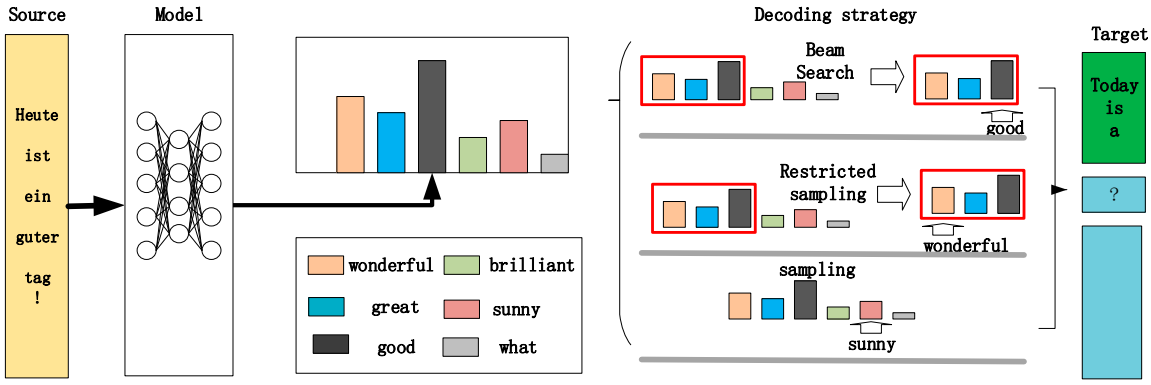

Figure 1. The difference of three decoder strategies: beam search, sampling, and restricted sampling.

For low-resource languages, the restricted sampling strategy gets a better result compared with the unrestricted [30], so we use the restricted sampling strategy in our decoding procedure.

3.3. Training Strategy

In our approach, we aim to generate diversified data based on the original data without any monolingual data. So, we use back-translation and self-learning on the initial training data to augment source or target data. We first train the two-directional model based on the idea of back-translation and self-learning, then use those models to decode the source or target sentence in the origin corpus. Finally, we combine multiple synthetic data with the original data to train the final model. The framework for our data augmentation method is presented in Figure 2.

The steps of our data diversification strategy are as follows:

Notations: Let S and T denote two languages, respectively. Let $D = (S, T)$ denote the bilingual training data set. We use R to represent the number of the training round. Let $M^R_{S\to T}$ denote the forward translation model at the Rth-round and $M^R_{T\to S}$ indicate the backward translation model at the Rth-round. We use $M^R_{l\to l',K}(X)$ to represent the results of dataset X translated by the model $M^R_{l\to l'}$, where K denotes a diversification factor.

Algorithm 1. Our data augmentation strategy.

Input: Parallel data $D = (S, T)$
  Diversification factor K
  Training round R
Output: $Model_{final}$
Procedure Forward ($D = (S,\ T)$)
  Initialize $M_{S\to T}$ with random parameters θ
  Train $M_{S\to T}$ on $D = (S, T)$ until convergence
Return $M_{S\to T}$
Procedure Backward ($D = (S,\ T)$)
  $D' = (T, S)$
  Initialize $M_{T\to S}$ with random parameters ϑ
  Train $M_{T\to S}$ on $D' = (T, S)$ until convergence
Return $M_{T\to S}$
Procedure DataDiverse ($D = (S, T), K, N$)
$D_0 \leftarrow D$
For r in R, $r \in 1, \ldots, N$:
do:
$D_r \leftarrow D_{r-1}$
$M^r_{S\to T}$←Forward (D_{r-1})
$M^r_{T\to S}$←Backward (D_{r-1})
For i in k do:
  T^i_r←Inference($M^r_{S\to T}$, S)
  S^i_r←Inference($M^r_{T\to S}$, T)
  D_r←$D_r \cup \left(S, T^i_r\right) \cup \left(S^i_r, T\right)$
$Model_{final}$←Train(D_R)
Return $Model_{final}$

In the first round, we train the backward NMT translation model $M^1_{T\to S}$ and forward NMT translation model $M^1_{S\to T}$ based on the initial dataset D^0. Then we employ the forward NMT model $M^1_{S\to T}$ to decode the source sentences of the training data by the restricted sampling strategy. We repeat the above processes to create multiple synthetic sentences on the target side. In other words, we gain numerous synthetic sentences as:

$$T^1_1 = M^1_{S\to T,1}(S), T^2_1 = M^1_{S\to T,2}(S), \ldots, T^K_1 = M^1_{S\to T,K}(S). \quad (5)$$

Similarly, we run the same process on the backward translation model $M^1_{T\to S}$. The multiple synthetic source sentences are generated by using the backward translation model $M^1_{T\to S}$ to translate the original target sentence.

$$S^1_1 = M^1_{T\to S,1}(T), S^2_1 = M^1_{T\to S,2}(T), \ldots, S^K_1 = M^1_{T\to S,K}(T), \tag{6}$$

Then, we add the multiple synthetic corpora to the original data as follows:

$$D^1 = (S,T) + \cup^K_{k=1}\left(S, T^k_1\right) + \cup^K_{k=1}\left(S^k_1, T\right). \tag{7}$$

If $R > 1$, we continue training the 2-round backward model $M^2_{T\to S}$ and forward model $M^2_{S\to T}$ based on D^1. We follow the process above until the final data set D^R is generated. Finally, we train the final translation model on corpus D^R. To aid understanding, Algorithm 1 summarizes this process.

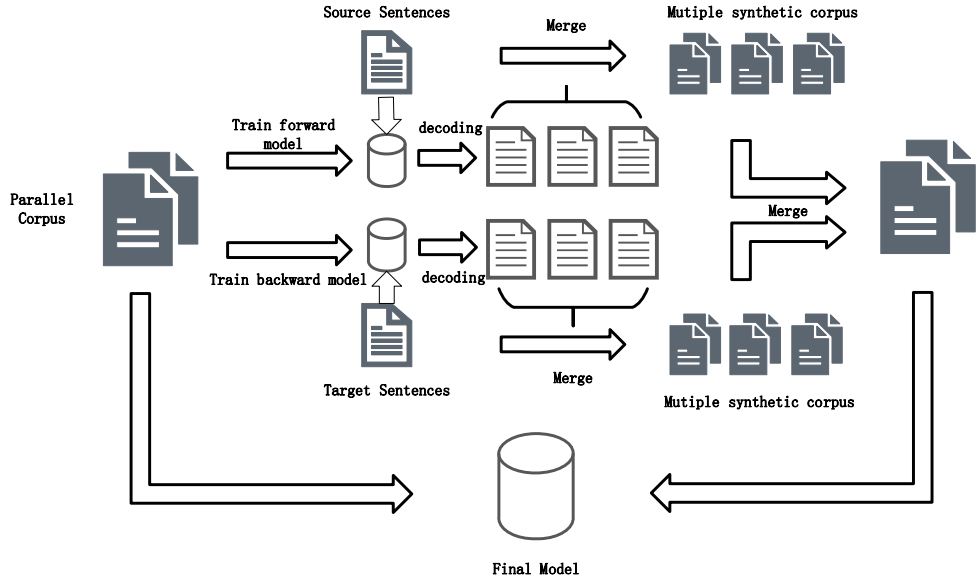

Figure 2. Overview of our data augmentation strategy.

4. Experiments

This section describes the experiments conducted to verify the effectiveness of our diversity data augmentation method. We first compared our method with other data augmentation methods on the IWSLT2014 English–German Translation tasks. This translation task can be used to simulate the low-resource setting. Then we verify the effectiveness of our method on three low-resource translation tasks.

4.1. IWSLT2014 EN–DE Translation Experiment

We used this dataset to simulate a low-resource setting. The dataset contains about 160k parallel sentences. We randomly sampled 5% of sentences from the training data as a validation set and concatenated IWSLT14.TED.dev2010 set, IWSLT.TED.dev2012 set, and three IWSLT14 test sets from 2010 to 2012 for testing. We used byte-pair-encoding (BPE) [16] to build a shared vocabulary of 10,000 tokens. All datasets were tokenized with the Moses toolkit [40]. We compared our method with other data augmentation methods mentioned in Zhu's work [26]. The data augmentation strategies were as follows:

- Artetxe et al. and Lample et al. propose to randomly swap words in nearby locations within a window size k; we denote it as SW [26,35,41].
- We denote the method that randomly drops words as DW [26,35,42].

- Xie et al. use a placeholder to replace word randomly. We denote this method as BW [9,26].
- Xie et al. employ a method that randomly replaces word tokens with a sample from the unigram frequency distribution over the vocabulary. We denote it as SmoothW [9,26].
- Kobayashi et al. randomly replace word tokens sampled from the output distribution of one language model [23,26]. We denote it as LMW.
- We denote Gao's work as SoftW. They randomly replace word embedding with a weight combination of multiple semantically similar words [26].

For model parameters, we used the basic parameter settings of the Transformer model. It consists of a 6-layer encoder and a 6-layer decoder. There are some exceptions: the model dimension was 512, the feed-forward dimension was 1024, and there were 4 attention heads. The dropout rate was 0.3, and the label smoothing was set as 0.1. We trained the models until convergence based on valid loss. At the decoding step, we set a beam size of 5 and a length penalty of 1.0. All the above data augmentation methods used the same setting as Zhu' s work [26]; we used the probability 0.15 to replace the word tokens in training steps. For our approach, unless specified otherwise, we used the same default setup, where $K = 3$ and $R = 1$.

Table 1 presents the results of the DE–EN translation task. As we can see, the DE–EN baseline based on the Transformer achieved 34.72 BLEU points without data augmentation. Compared with the baseline, we can find that our method substantially improved the translation performance by 1.96 BLEU points. Compared with other data augmentation methods, our approach obtained the best result. These results verify the effectiveness of our approach.

Table 1. The BLEU score of different data augmentation.

Model	DE–EN
Base	34.72
SW	34.70 *
DW	35.13 *
BW	35.37 *
SmoothW	35.45 *
LMW	35.40 *
SoftW	35.78 *
Our method	36.68

* denote the numbers are reported from Zhu' s work [23], other are based on our runs.

4.2. Low-Resource Translation Tasks

We also verified the effectiveness of three low-resource language translation tasks (i.e., English–Turkish (EN–TR), English–Nepali (EN–NE), and English–Sinhala (EN–SI) tasks). Both Nepali and Sinhala are very challenging languages to translate because their morphology and syntax are different compared with high-resource languages such as English. Meanwhile, there are not many users or parallel corpora, so data resources are particularly scarce.

Table 2 presents the statistics of three low-resource corpora. For the EN–TR experiment, we combined WMT EN–TR training sets and IWSLT14 training data, and the final corpus contained about 350k sentence pairs. We chose dev2010 and test2010 as the validation sets and used the four test sets from 2011 to 2014 as test sets. We learned the BPE vocabulary jointly on the source and target language sentences, and the vocabulary was built with 32k merge operations. The English–Sinhala training data contained about 400k pairs, while the English–Nepali training data had about 500k pairs. We used the same development set and test set as in Guzman' work [43]. We used the Indic NLP library [44] to tokenize the Nepali and Sinhala corpora. We used sentencepiece toolkits [45] to build the shared vocabulary. The size of the vocabulary was 5000.

Table 2. The statistics of the three low-resource corpora.

Model	Train	Dev	Test
TR–EN	0.35M	2.3k	4k
SI–EN	0.4M	2.9k	2.7k
NE–EN	0.56M	2.5k	2.8k

For the TR–EN translation task, we chose the base Transformer as our model structure. The model parameters were as follows: the number layers of the encoder was 6, the number of decoder layers was 6, the dimension of the inner feed-forward layer was 2048, the model dimensions was 512, and the number of attention heads was 8. The Adam optimizer was used, and the learning rate was 0.001, the dropout rate was 0.3, there were 4000 warm-up steps, and the models ran for 50 epochs. For inference, we set the beam size to 5 and the length penalty to 1.0, and averaged the last five checkpoints. We used the BLEU scores to measure the performance of the final model. For EN–NE and EN–SI translation tasks, we used a Transformer architecture with a 5-layer encoder and a 5-layer decoder, and each layer had two attention heads. The embedding dimension and feed-forward dimension were 512 and 2048. To regularize our models, we set a dropout rate of 0.4, label smoothing of 0.2, and weight decay of 10^{-4}. We set the batch size to 16,000 tokens and trained the model for 100 epochs. At the inference step, for NE–EN and SI–EN tasks, we used the length penalty of 1.2. We used the detokenized sacreBLEU [46] for these tasks.

Table 3 shows the TR–EN translation results of the different test sets. We can observe that our method was able to boost the translation quality compared with the baseline, which is without data diversification. We averaged the BLEU of four test sets, and our method achieved a 1.51 BLEU improvement. From Table 4, it can be seen that our method achieved a more than 1.0 BLEU improvement without using other monolingual data. These results indicate that augmenting the training data with diversity pseudo parallel data is useful for improving the translation performance.

Table 3. The BLEU scores of the TR–EN translation task.

Test	Test2011	Test2012	Test2013	Test2014	AVG
Baseline	24.08	24.79	26.38	25.03	
Our model	25.55	26.11	28.26	26.41	1.51

Table 4. The BLEU score of NE–EN and SI–EN translation tasks.

Model	NE–EN	SI–EN
Baseline	7.64	6.68
Our model	8.92	8.21

5. Discussion

In this section, we analyze the proposed data augmentation approach from the following perspectives: (1) Is the improvement of our method due to the multiple copies of the original data? (2) Which is more important between the backward model and forward model? (3) What effect does the different sampling values have on the performance?

5.1. Copying the Original Data

We copied the initial data seven times and merged the data to train the model in order to verify whether the performance of translation improved due to the increase of the data, denoted as 7Baseline. We ran two experiments based on two languages: EN–DE, TR–EN, and used the same parameter settings. Table 5 shows the BLEU scores of two data sets. We found that the model based on copied data consistently decreased the performance by 0.1 to 0.6 BLEU scores in all translation tasks. However,

our method yielded an improvement of 1.0 to 2.0 BLEU points on two datasets. These results verify that the increase in performance was not due to data replication, and that our approach can produce diverse data.

Table 5. Performance of copying original data multiple times.

Method	DE–EN	TR–EN (Test2013)
Baseline	34.72	26.38
7Baseline	34.51	26.23
Our method	36.68	28.26

Nguyen' work [15] showed that using random seeds to train translation models yields different model distributions. We trained three forward translation models and three backward translation models with a different random seed, then we used those models to translate training data to generate different synthetic corpora and combine them to train the final model. The parameters of those experiments were the same as our approach. Table 6 shows the results of the EN–DE translation task and EN–TR translation tasks. From Table 6, we observe that the approach based on random seed also yielded improvements of more than 2.01 and 1.89 in the BLEU scores for EN–DE and EN–TR tasks respectively, indicating the effectiveness of our data augmentation method. Our approach reached the same conclusion by using fewer translation models. Therefore, decoding with the restricted sampling increased the diversity of the training data, and our method could also further enhance translation performance.

Table 6. Comparison of our work with different random seeds.

Method	DE–EN	TR–EN (Test2013)
Baseline	34.72	26.38
Random	36.73	28.27
Our method	36.68	28.26

5.2. Backward Data or Forward Data

We perform experiments on EN–TR translation tasks, where we use the diversity data generated by the forward model and the backward model separately. We also compare these models with our bidirectional diversified model and the baseline model without data diversification. The results are shown in Table 7. We can observe that both backward and forward diversification models are still valid but worse than bidirectional diversification. We also find that diversification with backward models outperforms the ones with the forward models. Those findings strongly support us that our approach by leveraging both forward and backward diversification is helpful.

Table 7. Performance of backward data or forward data.

Method	Test2011	Test2014
	TR–EN	TR–EN
Baseline	24.08	25.03
Forward	24.42	25.07
Backward	25.08	25.94
Bidirectional	25.55	26.41

5.3. The Number of Samplings

We conducted experiments on the different synthetic data based on the number of samplings. We used the same translation models to translate source and target sentences with different decoding strategies, including unrestricted sampling from the model distribution, restricting sampling to the 5 or

10 highest-scoring outputs at every time step. Table 8 shows the BLEU scores for EN–DE and EN–TR translation tasks. From Table 8, we observe that restricted sampling outperformed the unrestricted sampling methods. Restricting sampling to the 5 highest-scoring outputs at every time step yielded the better result. This is because restricted sampling is the middle-ground between beam search and unrestricted sampling; it is unlikely to choose a lower-scoring output but still retains some randomness, especially in low-resource language translation tasks.

Table 8. Performance of decoding with the different sampling methods.

Method	DE–EN	TR–EN(Test2013)
Baseline	34.72	26.38
Sample_5	36.68	28.27
Sample_10	36.48	28.23
Sample	36.59	27.86

Sample_K represent that the model based on restricting sampling to the K-highest-scoring outputs at every time steps.

6. Conclusions

In this paper, we proposed a novel approach that is very effective in improving the performance of low-resource translation tasks. We trained a forward and backward model to translate the training data multiple times by restricted sampling, then used them to train the final model. The experimental results demonstrated that the proposed method was effective in many translation tasks. It outperformed the baselines in the IWSLT English–German translation task, IWSLT English–Turkish translation task, and two low-resource language translation tasks by 1.0–2.0 BLEU. Other experiments were conducted to analyze why the proposed method was effective. We found that our method could increase the data diversification of training data without extra monolingual data. Using bidirectional diversification methods is better than using either alone.

Author Contributions: Y.L., X.L., Y.Y. and R.D. conceived the approach and wrote the manuscript. All authors have read and approved the final manuscript.

Funding: This work is supported by the Open Project of Key Laboratory of Xinjiang Uygur Autonomous Region (2018D04018), the National Natural Science Foundation of China (U1703133), the Youth Innovation Promotion Association of the Chinese Academy of Sciences (2017472), and the High-level Talents Introduction Project of Xinjiang Uyghur Autonomous Region (Y839031201).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Neural Information Processing Systems 27 (NIPS 2014), Montreal, Canada, 8–13 December 2014; pp. 3104–3112.
2. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
3. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025.
4. Koehn, P.; Knowles, R. Six challenges for neural machine translation. *arXiv* **2017**, arXiv:1706.03872.
5. Zoph, B.; Yuret, D.; May, J.; Knight, K. Transfer learning for low-resource neural machine translation. *arXiv* **2016**, arXiv:1604.02201.
6. Gu, J.; Wang, Y.; Chen, Y.; Cho, K.; Li, V.O.K. Meta-learning for low-resource neural machine translation. *arXiv* **2018**, arXiv:1808.08437.
7. Ren, S.; Chen, W.; Liu, S.; Li, M.; Zhou, M.; Ma, S. Triangular architecture for rare language translation. *arXiv* **2018**, arXiv:1805.04813.
8. Firat, O.; Cho, K.; Sankaran, B.; Vural, F.T.Y.; Bengio, Y. Multi-way, multilingual neural machine translation. *Comput. Speech Lang.* **2017**, *45*, 236–252. [CrossRef]

9. Xie, Z.; Wang, S.I.; Li, J.; Lévy, D.; Nie, A.; Jurafsky, D.; Ng, A.Y. Data noising as smoothing in neural network language models. *arXiv* **2017**, arXiv:1703.02573.
10. Gal, Y.; Ghahramani, Z. A theoretically grounded application of dropout in recurrent neural networks. In Proceedings of the Advances in Neural Information Processing Systems 29 (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 1019–1027.
11. Sennrich, R.; Haddow, B.; Birch, A. Improving neural machine translation models with monolingual data. *arXiv* **2015**, arXiv:1511.06709.
12. Zhang, J.; Zong, C. Exploiting Source-side Monolingual Data in Neural Machine Translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016.
13. Ueffing, N.; Haffari, G.; Sarkar, A. Transductive learning for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 23–30 June 2007; pp. 25–32.
14. Imamura, K.; Fujita, A.; Sumita, E. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, Melbourne, Australia, 15–20 July 2018; pp. 55–63.
15. Nguyen, X.P.; Joty, S.; Kui, W.; Aw, A.T. Data Diversification: An Elegant Strategy For Neural Machine Translation. *arXiv* **2019**, arXiv:1911.01986.
16. Sennrich, R.; Haddow, B.; Birch, A. Neural machine translation of rare words with subword units. *arXiv* **2015**, arXiv:1508.07909.
17. Pan, Y.; Li, X.; Yang, Y.; Dong, R. Morphological Word Segmentation on Agglutinative Languages for Neural Machine Translation. *arXiv* **2020**, arXiv:2001.01589.
18. Sennrich, R.; Haddow, B. Linguistic input features improve neural machine translation. *arXiv* **2016**, arXiv:1606.02892.
19. Tamchyna, A.; Marco, M.W.D.; Fraser, A. Modeling target-side inflection in neural machine translation. *arXiv* **2017**, arXiv:1707.06012.
20. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. *arXiv* **2017**, arXiv:1608.06993.
21. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
22. Fadaee, M.; Bisazza, A.; Monz, C. Data augmentation for low-resource neural machine translation. *arXiv* **2017**, arXiv:1705.00440.
23. Kobayashi, S. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv* **2018**, arXiv:1805.06201.
24. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
25. Wu, X.; Lv, S.; Zang, L.; Han, J.; Hu, S. Conditional BERT contextual augmentation. In Proceedings of the International Conference on Computational Science, Faro, Portugal, 12–14 June 2019; pp. 84–95.
26. Gao, F.; Zhu, J.; Wu, L.; Xia, Y.; Qin, T.; Cheng, X.; Zhou, W.; Liu, T. Soft Contextual Data Augmentation for Neural Machine Translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5539–5544.
27. Poncelas, A.; Shterionov, D.; Way, A.; de Buy Wenniger, G.M.; Passban, P. Investigating backtranslation in neural machine translation. *arXiv* **2018**, arXiv:1804.06189.
28. Burlot, F.; Yvon, F. Using monolingual data in neural machine translation: A systematic study. *arXiv* **2019**, arXiv:1903.11437.
29. Cotterell, R.; Kreutzer, J. Explaining and generalizing back-translation through wake-sleep. *arXiv* **2018**, arXiv:1806.04402.
30. Edunov, S.; Ott, M.; Auli, M.; Grangier, D. Understanding back-translation at scale. *arXiv* **2018**, arXiv:1808.09381.
31. Currey, A.; Miceli-Barone, A.V.; Heafield, K. Copied monolingual data improves low-resource neural machine translation. In Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2017; pp. 148–156.
32. Cheng, Y. *Semi-Supervised Learning for Neural Machine Translation. Joint Training for Neural Machine Translation*; Springer: Cham, Switzerland, 2019; pp. 25–40.

33. He, D.; Xia, Y.; Qin, T.; Wang, L.; Yu, N.; Liu, T.Y.; Ma, W.Y. Dual learning for machine translation. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 9 December 2016; pp. 820–828.
34. Lample, G.; Conneau, A.; Denoyer, L.; Ranzato, M. Unsupervised machine translation using monolingual corpora only. *arXiv* **2017**, arXiv:1711.00043.
35. Lample, G.; Ott, M.; Conneau, A.; Denoyer, L.; Ranzato, M. Phrase-based & neural unsupervised machine translation. *arXiv* **2018**, arXiv:1804.07755.
36. Hoang, V.C.D.; Koehn, P.; Haffari, G.; Cohn, T. Iterative back-translation for neural machine translation. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, Melbourne, Australia, 15–20 July 2018; pp. 18–24.
37. Niu, X.; Denkowski, M.; Carpuat, M. Bi-directional neural machine translation with synthetic parallel data. *arXiv* **2018**, arXiv:1805.11213.
38. Zhang, J.; Matsumoto, T. Corpus Augmentation by Sentence Segmentation for Low-Resource Neural Machine Translation. *arXiv* **2019**, arXiv:1905.08945.
39. Ott, M.; Auli, M.; Grangier, D.; Ranzato, M. Analyzing uncertainty in neural machine translation. *arXiv* **2018**, arXiv:1803.00047.
40. Koehn, P.; Och, F.J.; Marcu, D. Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Edmonton, AB, Canada, 27 May–1 June 2003; pp. 48–54.
41. Artetxe, M.; Labaka, G.; Agirre, E.; Cho, K. Unsupervised neural machine translation. *arXiv* **2017**, arXiv:1710.11041.
42. Iyyer, M.; Manjunatha, V.; Boyd-Graber, J.; Daumé III, H. Deep unordered composition rivals syntactic methods for text classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 1681–1691.
43. Guzmán, F.; Chen, P.J.; Ott, M.; Pino, J.; Lample, G.; Koehn, P.; Chaudhary, V.; Ranzato, M. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv* **2019**, arXiv:1902.01382.
44. Indic NLP Library. Available online: https://github.com/anoopkunchukuttan/indic_nlp_library (accessed on 5 May 2020).
45. SentencePiece. Available online: https://github.com/google/sentencepiece (accessed on 5 May 2020).
46. Post, M. A call for clarity in reporting BLEU scores. *arXiv* **2018**, arXiv:1804.08771.