Generating Diverse Back-Translations via Constraint Random Decoding

Yiqi Tong[1,2,3], Yidong Chen[1,2*], Guocheng Zhang[1,2], Jiangbin Zheng[1,2,4], and Hongkang Zhu[1,2] and Xiaodong Shi[1,2]

[1] Department of Artificial Intelligence, School of Informatics, Xiamen University
[2] Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan, Ministry of Culture and Tourism
[3] Institute of Artificial Intelligence, Beihang University
[4] AI Lab, School of Engineering, Westlake University
{yqtong,zgccc,hkzhu}@stu.xmu.edu.cn
{ydchen,mandel}@xmu.edu.cn
zhengjiangbin@westlake.edu.cn

Abstract. Back-translation has been proven to be an effective data augmentation method that translates target monolingual data into source-side to improve the performance of Neural Machine Translation (NMT), especially in low-resource scenarios. Previous researches show that diversity of the synthetic source sentences is essential for back-translation. However, the frequently used random methods such as sampling or noised beam search, although can output diverse back-translations, often generate noisy synthetic sentences. To alleviate this problem, we propose a simple but effective constraint random decoding method for back-translation. The proposed method is based on an automatic post-editing (APE) data augment framework, which incorporates fluency boost learning. Moreover, to increase the diversity of synthetic data and ensure quality, we proposed to use an evolution decoding algorithm. Compared with the original back-translation, our method can generate more diverse while less noisy synthetic sentences. The experimental results show that the proposed method can get 0.6 BLEU improvements on the WMT18 EN-DE news dataset and more than 0.4 BLEU improvements on the EN-ZH dataset which is in the medical field, respectively.

Keywords: NMT · Back-translation · Automatic post-editing · Evolution decoding algorithm.

1 Introduction

In the past years, attention-based Neural Machine Translation (NMT) has become the mainstream approach because of its significant performance [1, 21, 20]. However, to achieve promising performance for a single language pair, millions of parallel sentences are necessary, which are data-hungry in many language pairs. To cope with this issue, researchers investigated using monolingual data for NMT and other natural language processing (NLP) tasks [2]. Specially, [15] proposed back-translation, which makes

[*] Corresponding author.

use of an NMT model with opposite translation direction to translate the target-side monolingual data into the source-side to enrich the parallel training corpus. However, the traditional back-translation still has problems. Current strong NMT model such as Transformer [21] adopts beam search in the decoding stage and generates candidates that only differ with one another by punctuation or minor morphological variations, making the translated sentences lack of diversity [12, 7]. On the other hand, the common alternative methods based on random decoding, such as sampling [26], often put too much noise into synthetic sentences, which reduce the data quality.

There are some works attempting to get more diverse and high-quality translation results, e.g. mirror-generative neural machine translation (MGNMT) [29], diverse beam search [22], adding an additional penalization term to expansion the same parent node [12], introducing a discrete latent variables to control generation [7, 16], manipulating attention heads [19], etc. While most of these studies have exploited decoding strategy, a few of them have tried developing automatic post-editing (APE) method to efficiently use the monolingual data.

In this work, we proposed a simple but effective constraint random decoding method for back-translation, which follows an APE framework. First, we build fluency boost sentence-pairs by combining the golden source-side sentences and the corresponding pseudo source-side sentences generated by back-translation. Then a sequence-to-sequence APE model was trained to re-generate pseudo source-side sentences, which will be used in the next iterations. Please note that, the above-mentioned process will be iterated several times.

Finally, we build synthetic fluency boost corpus by combining the source-side fluency boost sentences which generated by APE and target-side golden sentences for data augment. During the APE decoding process, a evolution decoding algorithm could be optionally adopted. Our methods can double the training data at maximum and can be applied to any encoder-decoder framework. As far as we know, we are the first to introduce fluency boost learning into the field of back-translation. Experimental results show that the proposed method can get 0.6, 0.4 BLEU improvements over the baseline model on EN-DE, EN-ZH test set.

2 Related work

The NMT system is known to be extremely data-needed. Previous works proved that the diversity of the training data can provide more discriminative information for the NMT model [5, 6]. However, high-quality parallel corpus is limited. To address above problem, [15] proposed back-translation, which utilize abundant amount of mono-lingual data during the model training process. [3, 25] broadens the understanding of back-translation and investigated a number of methods like unrestricted sampling, large-scale noised training to generate synthetic source sentences. To explore the actual effects of the back-translation, [14] studied the performance of EN-DE NMT models when incrementally larger amounts of synthetic data are used for training.

Some recent works have looked at the diverse decoding method for NMT. [22] proposed diverse beam search that modifies classical beam search algorithm with a diversity augmented sequence decoding objective and get state-of-the-art results on several language generation tasks at that time. Other than design diversity encouraging decoding algorithm, [7, 16] proposed mixture model, which could improve both quality and diversity of the translations by introduced latent variables to control generation. However, this method will increase the difficulty of model training [19]. More recently, to make better use of non-parallel data, [29] proposed a mirror-generative NMT model (MGNMT), which outperforms previous approaches in all investigated scenarios by combing the source-side and target-side monolinguals and corresponding language models organically during the training phase.

Our work was partly followed with [4], which they proposed a fluency boost learning and inference mechanism and get significant improvement over the former Grammar Error Correction (GEC) models. However, they focused on generating more error-corrected data, while we use this strategy to iteratively enhance the predictions of back-translation by rewriting the sentences with our proposed APE model and providing more training signals for NMT model. Moreover, we also incorporated a novel evolution decoding algorithm in the model decoding stage to get more diverse candidates.

3 Proposed Methods

3.1 Fluency Boost Learning

Fluency boost learning (FBL) is an iterative learning strategy, which was first proposed by [4] for solving the GEC problem [27, 9]. GEC aims for automatically correcting various types of errors in the given text, while there are mainly Rule-based approaches [17], MT-based approaches [13] and LM-based [18] to solve this problem.

In this paper, we transfer it to the field of back-translation and proposed an Automatic Post-Editing (APE) model which enable to learn how to improve a sentence's diversity and quality without changing its original meaning by FBL. Specifically, the sentences generated by back-translation usually have various errors. Hence we treat it as a MT-based GEC problem, which the source-side is pseudo sentences generated by back-translation and the target-side is golden sentence from parallel corpus. Figure 1 illustrates the training process of our APE model, where PD is parallel dataset, MD is monolingual dataset, NMT and APE are neural machine translation model and automatic post-editing model, respectively. Superscript $\#$ denotes the machine translation results, subscript src and trg are source-side sentences and target-side sentences respectively, P stands for it generated by monolingual data. Specifically, We use parallel corpus $PD_{src-trg}$ to training the back-translation model $NMT_{trg-src}$, then we can obtain the fluency boost sentence pairs $PD_{src-src\#}$ by combining MD^{p}_{src} and $MD^{P}_{src\#}$, Where the former is obtained by $PD_{src-trg}$ and the latter is generated from $NMT_{trg-src}$ by decoding MD^{D}_{src}. Finally, we use $PD_{src-src\#}$ to training the NMT-based APE model.

The aim of the NMT model is to maximize the probabilities P of the target languages $\mathbf{Y} = (y_1, ..., y_j)$ given the source language sequences $\mathbf{X} = (x_1, ..., x_i)$, which calculated as follows:

$$argmax \frac{1}{N} \sum_{n=1}^{N} log(P_\theta(\mathbf{Y}^n|\mathbf{X}^n) \tag{1}$$

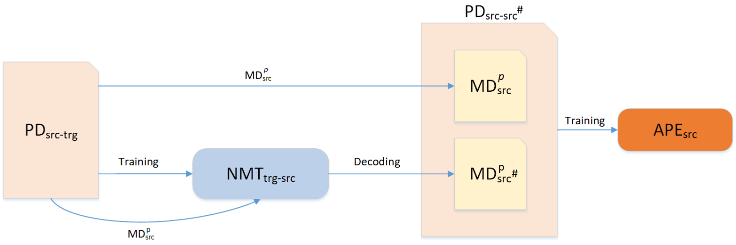

Fig. 1. The training process of our APE model.

Where n is the n-th sentence in corpus with a total number of N and θ is model parameters.

In this work, the transformer architecture is used for both NMT and APE models. The difference is, when apply to APE model, the target-side is the sentences which contain various grammatical errors. Our method can be applied to any encoder-decoder framework without any code changes. We expect that other neural sequence-to-sequence based generative model could benefit from our approach, but the choice of the model architecture is not a focus of this paper.

3.2 Evolution Decoding Algorithm

Beam search is a limited-width breadth first search algorithm [11]. For a input sentence x, the generated candidate sequences $\{y_1, ..., y_j\}$ by beam search are highly similar, especially at the beginning part, which is harmful for back-translation or our APE model to generate diversity data. The evolution algorithm [8] is inspired by Darwin's theory, which simulates the natural evolution process of gene sequence and make the next generation of genes stronger through of fittest. When in the decoding stage of back-translation, N-best candidates can be generated like a set of gene sequences. Some words between these candidates are different but have similar semantics, which just provides the basic pre-conditions for our evolution decoding algorithm.

For this part, to further increase the diversity of synthetic data and ensure the quality as much as possible, we proposed an evolution decoding algorithm (EDA), which summarized in Algorithm 1. Formally, we use the offline method[1] to integrate our algorithm into the training process.

As shown in Algorithm 1, EDA selects m-best candidate sequences as the initial population and uses crossover and mutation to modify the original sequence to achieve diversity improvement. For crossover operation, we exchange fragments of two adjacent candidate sequences. If db_i and db_j were chosen, we simply split db_i and db_j into s_i^0 and s_i^1, s_j^0 and s_j^1 from the middle position, where superscript 0/1 is the first/second half of the candidate sequence. When using beam search decoding, as we mentioned above, s_i^0 and s_i^1 are usually same, so we set 10% probability to exchange s_i^0 and s_j^0,

[1] We still use beam search at the model training stage but use EDA at the decoding stage of back-translation or APE.

Algorithm 1 Evolution decoding algorithm

```
Input: Beam search decoding sequence $DB$, beam size $b$, batch size $n$, the sample size $m$,
    maximum number of iterations $T$ and the number of wining samples $k$
Output: Wining sequence set $DB^*$
Parameters initial: Population init_pop = $DB^0$, time step $t$ = 0, the samples of the wining
    sequences are initialized as candidate sequences $db_i^*$ = $db_i$ and initial diversity $D^0$, where $D_j^0$
    is the diversity score of the sample $j$ at time step 0
do
  for each wining sequence $db_{j_t}^*$ do
    $db_{j_{t+1}}$ ⇐ crossover($db_{j_t}^*$) # Crossover
    $db_{j_{t+1}}^{'}$ ⇐ mutate($db_{j_{t+1}}$) # Mutation
    $D_j^{t+1}$ ⇐ fitness_function($db_{j_{t+1}}^{'}$) # Calculate the fitness of sample candidates
    $db_{j_{t+1}}^*$ ⇐ select($D_j^{t+1}$, $db_{j_{t+1}}^{'}$) # Update winning sequence
    $D^{t+1}$ += $\frac{1}{n}D_j^{t+1}$ # Calculate total fitness
    $DB^{t+1}$ ⇐ update($DB^t$, $db_{j_{t+1}}^*$) # Update wining sequence set
  $t = t + 1$
while $D^{t+1} > D^t$ and $t + 1 < T$
return $DB^*$
```

90% probability to exchange s_i^1 and s_j^1. Although this may cause the newly generated sequence to be incoherent, previous works [23, 24] proved that adding noise to the source-side data can make the model more robust. To avoid getting too much noise, the training data choose 15% of the sequences at random for mutation operation, which follows [2]. If the i-th sequence is chosen, we random replace i-th word with a random word from the vocabulary. We constructed a fitness function to measure sequence diversity, which calculated as follows.

$$d_n = \frac{m\ of\ unique\ n - grams\ in\ k\ translations}{total\ m\ of\ n - grams\ in\ k\ translations} \tag{2}$$

$$u_i = \sum_{i=1}^{k} \frac{unique(s_j) - same(s_j, s_i)}{len(s_j)} \tag{3}$$

$$D_j = \sum_{n=1}^{N} d_j^n + u_j \tag{4}$$

where D_j is our final diversity score for sequence j, which was calculated by d_n and u_i. Specifically, d_n reflects the degree of sub-sequence repeatability for a given sequence, Which higher score means it contains more unique n-gram tuples. We set n to 2 in experiments. However, for a too short sequence, d_n may give an overly high evaluation. To address this shortcoming, we adapt another diversity metric u_j, which measures the difference between j-th sequence and others. Where function $unique()$ counts unique words of s_j and $same()$ calculates the identical words between s_j and s_i.

In theory, for a input sequence x, we can get $k - 1$ candidates by EDA. After the t-th iteration, $(k - 1)2^{(t-1)} + 1$ samples will be generated. To reduce the pressure of

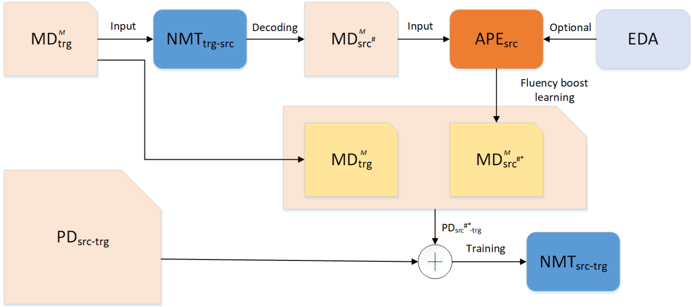

Fig. 2. The whole training process of our NMT model based on FBL and EDA.

computation and memory, we only keep the top-k sequences when making the selection operation.

3.3 Joint Training

Figure 2 illustrates the overall architecture of our proposed methods. Where PD is parallel dataset, MD is monolingual dataset, superscript $\#$ and $*$ denotes the synthetic data that generated by NMT and APE respectively, M stands for it comes from bilingual dataset. Our final goal is to get the diversity and high-quality sentence-pairs to improve the performance of NMT model. Therefore, in the first step, $NMT_{trg-src}$ which is trained by golden parallel data $PD_{src-trg}$ to back-translate the target-side monolingual data MD_{trg}^{M} into the source-side pseudo monolingual data $MD_{src\#}^{M}$, then we training the APE_{src} to boost the fluency of $MD_{src\#}^{M}$ and get the fluency-boosted monolingual data $MD_{src\#*}^{M}$. We can carry out multiple rounds of APE to gradually improve the fluency of the corpus. And EDA was applied optional during the APE decoding stage. Finally, we merge $PD_{src-trg}$ and synthetic parallel corpus $PD_{src-trg}^{*}$ to training the $NMT_{src-trg}$.

4 Experimental Setting

4.1 Metrics

To quantitatively assess the quality and diversity of the translation results, we use perplexity to measure the fluency of translation sentences, which a lower perplexity score means the better generalization performance. For evaluate the overall performance of the NMT model, standard BLEU score was calculated. And we use DEQ (Diversity Enhancement per Quality, [19]) to measure the diversity and quality, which was calculated as follows.

$$DEQ = \frac{(pwb^{*} - pwb)}{(ref^{*} - ref)} \tag{5}$$

Where rfb and rfb^{*} refer to reference BLEU score of the evaluated system and baseline respectively, pwb and pwb^{*} refer to pair-wise BLEU score of the evaluated system and baseline respectively, which was calculated as follows.

$$pwb = BLEU([y^j], y^k)_{j \in [k], k \in [k], j \neq k} \tag{6}$$

Where $\{y^1, y^2, ..., y^k\}$ are k translation hypotheses of a source sentence x. Lower pwb and higher rfb means better results.

4.2 Dataset

We evaluate NMT training on parallel corpus and with additional monolingual data, which consist of the following five parts.

2M EN-DE. We randomly select 2M sentence-pairs in the news filed from WMT18 for English-German translation task.

80K EN-DE. To simulate low-resource scenarios, we randomly select 80K sentence-pairs from 2M EN-DE.

2M EN-ZH. We randomly select 2M sentence-pairs in the medical field from 10M English-Chinese which collected by our own.

2M DE. Contain 2M German monolingual sentences from News Crawl.

2M ZH. Contain 2M Chinese monolingual sentences from 10M EN-ZH, which the 2M ZH training data has been excluded.

Finally, we choose newstest2013-2018 and randomly select 3K from 10M EN-ZH as our test set for EN-DE task and EN-ZH task respectively, which 2M EN-ZH training data has already been excluded.

4.3 Experiment Settings

We use the Moses tokenizer [10] and learn a joint source and target Byte-Pair-Encoding [15] by fastBPE [2] with 35K types. Before we conduct the random selection, all sentences were lowercased, and which length longer than 150 sub-words were removed. We also remove the sentence pairs whose length ration exceed 1.5 between the source-side and the target-side. The hyper-parameters for our neural NMT and APE model are adopt from [28]. All models are trained on NVIDIA GeForce RTX 2080Ti GPUs and use label smoothing with a uniform prior distribution over the vocabulary $\epsilon = 0.1$. We use same hyper-parameters for all experiments.

5 Results and Analysis

5.1 Main Results

As shown in Table 1, We conducted experiments on two different data scale. Where BITEXT is baseline NMT system without adopting any data augment methods. +BEAM and +SAMPLING are NMT systems with standard back-translation, which adopts different decoding strategies. +APE are our proposed APE model with different iterations. As the APE rounds increase, the BLEU score shows an overall upward trend, which is

[2] https://github.com/glample/fastBPE

Table 1. Under different data scenarios, model performance comparison.

80k bilingual training data and with 2M monolingual data							
Models	NST13	NST14	NST15	NST16	NST17	NST18	AVG
BITEXT	15.25	14.32	16.51	19.09	15.99	21.05	17.03
BITEXT+BEAM	21.47	22.26	23.85	28.41	23.52	32.19	25.28
BITEXT+SAMPLING	21.38	21.85	23.86	28.81	23.21	32.16	25.21
BITEXT+APE1	21.44	22.48	**24.56**	**29.43**	23.73	32.89	25.76
BITEXT+APE2	21.51	22.71	24.27	29.06	23.73	**33.57**	25.81
BITEXT+APE3	**21.66**	**22.86**	24.55	29.27	**23.78**	33.31	**25.90**
2M bilingual training data and with 2M monolingual data							
Models	NST13	NST14	NST15	NST16	NST17	NST18	AVG
BITEXT	24.64	25.83	27.93	33.82	27.49	39.76	29.91
BITEXT+BEAM	25.94	27.76	29.6	35.77	28.76	42.12	31.66
BITEXT+SAMPLING	25.58	27.59	29.86	35.85	28.91	42.35	31.69
BITEXT+APE1	**26.15**	**28.41**	**30.22**	36.17	29.11	42.61	32.11
BITEXT+APE2	26.13	27.98	30.05	**36.31**	**29.16**	43.2	**32.14**
BITEXT+APE3	25.95	28.15	30.06	36.12	28.87	**42.75**	31.98

Table 2. The comparison of different data augment methods.

Methods	perplexity	pwb	|DEQ|
Baseline	478.00	75.68	0
Back-translation	392.11	80.01	1.89
APE	**324.35**	**73.02**	**3.11**

consistent with our assumption, with the iteration of APE, there will be more higher-quality candidate translations to choose from. Compared with +BEAM, our best model can achieve average 0.60 and 0.48 BLEU score improvement respectively through APE. These results suggest that our method can improve the quality of back-translation. Moreover, insufficient NMT model training leads to the poor-quality of back-translation. So our method is more effective in low-resource scenarios, which the improvement of 80K BITEXT+APE is larger than 2M BITEXT+APE.

5.2 Quantitative Analysis

Furthermore, we want to prove that our proposed method can both improve diversity and quality of the synthetic data. For this purpose, we conducted analysis experiments on the translations and use perplexity, Pair-wise BLEU (pwb) and DEQ as evaluation metrics.

Specifically, to evaluate the APE model at corpus level, we randomly select 7M monolingual data from News Crawl to training a 5-gram language model, then use it to calculate the average perplexity. And we select 5-best candidates for each source sentences in newstest2013-2018 to calculate pwb and DEQ score. As shown in Table 2, through our proposed fluency boost learning method, the quality and diversity of synthetic data are significantly improved. Compared with the baseline system BITEXT, the perplexity and pwb dropped by 153.65 and 2.66 through APE, which indicating that both the diversity and quality of the translations have been improved. On the contrary,

Table 3. Model performance comparison between different decoding strategies.

Decoding Strategy	NST13	NST14	NST15	NST16	NST17	NST18	AVG
BEAM	25.94	27.76	29.6	35.77	28.76	42.12	31.66
SAMPLING	25.58	27.59	29.86	35.85	28.91	42.35	31.69
EDA_N-GRAM	25.86	**28.14**	29.96	36.05	**29.14**	42.54	31.95
EDA_DIFF	25.54	27.3	29.98	35.74	28.88	42.31	31.63
EDA	25.72	27.51	30.1	35.94	29.01	42.6	31.81
APE+EDA	**26.02**	28.01	**30.15**	**36.14**	28.83	**43.02**	**32.03**

Table 4. Translation diversity and quality comparison.

Decoding strategy	perplexity	pwb
Beam search	364.06	80.01
SAMPLING	1138.28	12.93
EDA	**418.78**	**74.79**

back-translation will both reduce the diversity and quality of the translations, which perplexity and pwb increased 67.76 and 1.22 respectively in our experiment. Finally, compared with back-translation, DEQ increased by 1.22, which also proves the diversity and quality are both improved by APE.

To further boost synthetic data diversity and explore the effectiveness of EDA, we conduct experiments to compare the performance of the NMT model with different decoding strategies. As shown in Table 3, all models are trained with 2M EN-DE and adopt 2M DE for data augment. Where EDA_DIFF adopts formula (2) as fitness function, EDA_GRAM and EDA adopts both formula (3) and (4) as fitness function, respectively. The average overall score of EDA is slightly higher than BEAM and SAMPLING system. With one round of fluency boosting, APE+EDA model achieved best performance. As mentioned in formula (2) and (3), we have defined two indicators to measure diversity for candidates re-ranking, so we perform ablation experiments to test the effects of the two indicators separately. The experimental result in Table 3 shows that the BLEU of EA_N-GRAM system is 0.29 higher than BEAM, indicating that it is feasible to use the diversity of sub-sequences to measure the diversity of whole sequence. On the other hand, EA_DIFF can also produce equivalent results to the standard beam search, but the effect is not as good as the EDA or EA_N-GRAM.

We also did a quantitative analysis of EDA, as shown in Table 4, our evolutionary decoding algorithm can achieve a compromise between beam search and sampling. Compared with beam search, EDA improves diversity of generated data. And compared with the sampling, EDA introduces less noise.

5.3 Qualitative Analysis

For testing the applicability of our propsoed model in other domains, we conducted experiments on 2M EN-ZH medical data and with 2M ZH monolingual data. As shown in Table 5, we can get conclusions similar to 2M EN-DE experiments, BITEXT+APE1 still get best performance. But the improvement brought by data augmentation is limited.

Table 5. Model performance on EN-ZH test set.

Model	Test set
BITEXT	37.23
BITEXT+BEAM	38.51
BITEXT+SAMPLING	37.98
BITEXT+APE1	**38.93**

Table 6. Case study.

Example 1	
Src	doch in amerika wird mehr so viel getanktwie früher .
Ref	americans don 't buy as much as gas as they used to .
Baseline	but in America ***is*** not a lot more than is before .
Ours	but in America, ***however***, is not much more the before .
Example 2	
Src	30 vorschläge standen zur auswah , fünf sind noch im rennen .
Ref	there were 30 proposals to choose from , five of which are still in the running .
Baseline	30 proposals were made ***to selection*** , five are still in the race .
Ours	***thirty*** proposals were made ***to select*** , five are still in the race .

We believe that the quality of 2M EN-ZH is higher than 2M EN-DE. So improvement brought by data augmentation is limited.

To observe the effect more intuitively, we give two examples to illustrate the improvement brought by our model. As shown in Table 6, all models are trained with 80K EN-DE and use 2M DE for data augment. Compared with baseline model, our model could not only correct word errors and grammatical errors like “is” in Example 1 but also improved sentences diversity like “however” in Example 1 and “thirty” in Example 2, which proved that our method can both improve the diversity and quality of the back-translation sentence by introducing fluency boost learning. However, due to the NMT model did not correctly translate “getankt” into “gas”, our model was not corrected it either.

6 Conclusion

To promote the diversity and quality of synthetic data generated by back-translation, in this paper, we proposed a fluency boost learning based data augment framework, which could extend the origin corpus and applied to any sequence to sequence machine translation model. Furthermore, we performed experiments on different language pairs and resource scenarios to prove our methods could boost both the quality and diversity of the synthetic corpus generated by back-translation. Finally, the experiment results on EN-DE and EN-ZH showed that our proposed methods were effective. In future work, we will explore the influence of noise bring by back-translation under different data scales and further improve our evolution decoding algorithm.

Acknowledgements

The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 62076211, U1908216 and 61573294, and the Outstanding Achievement Late Fund of the State Language Commission of China under Grant WT135-38.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
3. Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. arXiv preprint arXiv:1808.09381 (2018)
4. Ge, T., Wei, F., Zhou, M.: Fluency boost learning and inference for neural grammatical error correction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1055–1065 (2018)
5. Gimpel, K., Batra, D., Dyer, C., Shakhnarovich, G.: A systematic exploration of diversity in machine translation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1100–1111 (2013)
6. Gong, Z., Zhong, P., Hu, W.: Diversity in machine learning. IEEE Access **7**, 64323–64350 (2019)
7. He, X., Haffari, G., Norouzi, M.: Sequence to sequence mixture model for diverse machine translation. arXiv preprint arXiv:1810.07391 (2018)
8. Hinterding, R., Michalewicz, Z., Eiben, A.E.: Adaptation in evolutionary computation: A survey. In: Proceedings of 1997 Ieee International Conference on Evolutionary Computation (Icec'97). pp. 65–69. IEEE (1997)
9. Junczys-Dowmunt, M., Grundkiewicz, R., Guha, S., Heafield, K.: Approaching neural grammatical error correction as a low-resource machine translation task. arXiv preprint arXiv:1804.05940 (2018)
10. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions. pp. 177–180. Association for Computational Linguistics (2007)
11. Kool, W., Van Hoof, H., Welling, M.: Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. arXiv preprint arXiv:1903.06059 (2019)
12. Li, J., Monroe, W., Jurafsky, D.: A simple, fast diverse decoding algorithm for neural generation. arXiv preprint arXiv:1611.08562 (2016)
13. Malmi, E., Krause, S., Rothe, S., Mirylenka, D., Severyn, A.: Encode, tag, realize: High-precision text editing. arXiv preprint arXiv:1909.01187 (2019)
14. Poncelas, A., Shterionov, D., Way, A., de Buy Wenniger, G.M., Passban, P.: Investigating backtranslation in neural machine translation (2018)
15. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709 (2015)
16. Shen, T., Ott, M., Auli, M., Ranzato, M.: Mixture models for diverse machine translation: Tricks of the trade. arXiv preprint arXiv:1902.07816 (2019)

17. Sidorov, G., Gupta, A., Tozer, M., Catala, D., Catena, A., Fuentes, S.: Rule-based system for automatic grammar correction using syntactic n-grams for english language learning (l2). In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task. pp. 96–101 (2013)
18. Stahlberg, F., Bryant, C., Byrne, B.: Neural grammatical error correction with finite state transducers. arXiv preprint arXiv:1903.10625 (2019)
19. Sun, Z., Huang, S., Wei, H.R., Dai, X., Chen, J.: Generating diverse translation by manipulating multi-head attention. In: AAAI. pp. 8976–8983 (2020)
20. Tong, Y., Zheng, J., Zhu, H., Chen, Y., Shi, X.: A document-level neural machine translation model with dynamic caching guided by theme-rheme information. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 4385–4395 (2020)
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
22. Vijayakumar, A.K., Cogswell, M., Selvaraju, R.R., Sun, Q., Lee, S., Crandall, D., Batra, D.: Diverse beam search for improved description of complex scenes. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
23. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. pp. 1096–1103 (2008)
24. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., Bottou, L.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research **11**(12) (2010)
25. Wu, L., Wang, Y., Xia, Y., Tao, Q., Lai, J., Liu, T.Y.: Exploiting monolingual data at scale for neural machine translation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 4198–4207 (2019)
26. Xu, W., Niu, X., Carpuat, M.: Differentiable sampling with flexible reference word order for neural machine translation. arXiv preprint arXiv:1904.04079 (2019)
27. Yannakoudakis, H., Rei, M., Andersen, Ø.E., Yuan, Z.: Neural sequence-labelling models for grammatical error correction. In: Proceedings of the 2017 conference on empirical methods in natural language processing. pp. 2795–2806 (2017)
28. Zhang, J., Ding, Y., Shen, S., Cheng, Y., Sun, M., Luan, H., Liu, Y.: Thumt: An open source toolkit for neural machine translation. arXiv preprint arXiv:1706.06415 (2017)
29. Zheng, Z., Zhou, H., Huang, S., Li, L., Dai, X.Y., Chen, J.: Mirror-generative neural machine translation. In: International Conference on Learning Representations (2019)