electronics

MDPI

Article

Sentence Augmentation for Language Translation Using GPT-2

Ranto Sawai, Incheon Paik * and Ayato Kuwana

School of Computer Science and Engineering, The University of Aizu, Fukushima 965-8580, Japan; m5251141@u-aizu.ac.jp (R.S.); m5241101@u-aizu.ac.jp (A.K.)
* Correspondence: paikic@u-aizu.ac.jp

Abstract: Data augmentation has recently become an important method for improving performance in deep learning. It is also a significant issue in machine translation, and various innovations such as back-translation and noising have been made. In particular, current state-of-the-art model architectures such as BERT-fused or efficient data generation using the GPT model provide good inspiration to improve the translation performance. In this study, we propose the generation of additional data for neural machine translation (NMT) using a sentence generator by GPT-2 that produces similar characteristics to the original. BERT-fused architecture and back-translation are employed for the translation architecture. In our experiments, the model produced BLEU scores of 27.50 for tatoebaEn-Ja, 30.14 for WMT14En-De, and 24.12 for WMT18En-Ch.

Keywords: neural machine translation; back-translation; GPT; BERT-fused; data augmentation

check for updates

Citation: Sawai, R.; Paik, I.; Kuwana, A. Sentence Augmentation for Language Translation Using GPT-2. *Electronics* **2022**, *10*, 3082. https://doi.org/10.3390/electronics10243082

Academic Editor: Rui Pedro Lopes

Received: 29 October 2021
Accepted: 8 December 2021
Published: 10 December 2021

1. Introduction

Machine translation (MT) is an important field of natural language processing (NLP), and the issue of improving translation performance by data augmentation has attracted the interest of researchers. In particular, the back-translation approach [1] that adds the 226M Deutsche monolingual corpus to the WMT14En-De training dataset has obtained a BLEU score of 35.0. In addition, the studies that currently have the top three BLEU scores with the WMT14En-De dataset use back-translation. This also shows that adding monolingual data considerably improves the performance of neural machine translators. However, monolingual data that are used in back-translation must be similar to the original dataset because a nonsimilar dataset cannot be back-translated correctly. The preparation of a large amount of data similar to the original can be an obstacle in using back-translation.

In recent years, there has been a noticeable improvement in the performance of sentence generators such as GPT-2 [2] and GPT-3 [3]. These sentence generators are based on scaling law, they have a very large number of parameters, and they are trained on a very large amount of text data. In the future, when models that can generate infinitely many different types of sentences become available, it may be possible to incorporate all the knowledge about the sentences into the translator. Therefore, research on sentence generators and machine translators will be intertwined in the future.

In this paper, we investigate monolingual data generation using GPT-2 for improving MT. The concept of pretraining in GPT-2 with a large text dataset can be considered as relevant sentence sequences. In addition, we use the BERT-fused model, which has recently shown superior results in WMT14En-De translation.

The contribution of our research is that we show that the sentences generated by the sentence generator can increase the accuracy of the translation. Moreover, our study shows that the performance increases with the number of sentences added. These results highlight an interesting fact that the performance of a neural network translator can also be improved by a neural network sentence generator as well. In addition, because of its randomness, the sentence generator can theoretically generate an infinite number of different sentences, which may further help to improve performance.

2. Related Work

Noising is a common data augmentation technique for increasing the number of sentences. This method is also used in back-translation and has been shown to be efficient and successful. Noising is a method that does three things: word dropout [4], word shuffling [5], and word blanking [5]. Based on noising, several data augmentation methods have been discovered.

Syntax-aware data augmentation [6] can improve model robustness by using commonly used data augmentation word blanking, dropout, and the replacement of unimportant words. To measure the importance of words, this method uses the dependency tree to obtain the importance of words, thus moving away from obtaining the importance by frequency. This method achieved a score 26.5 in sacreBLEU in the WMT14En-De experiment.

Another approach to increase the text dataset is easy data augmentation [7]. This method uses synonym replacement, random deletion, random swap, and random insertion. In particular, synonym replacement and random insertion are methods of replacing or inserting words using synonyms. This method improved the performance of the recurrent neural network (RNN) and convolutional neural network (CNN).

The main difference between our study and data augmentation studies, such as noising, syntax-aware data augmentation, and easy data augmentation, is whether the amount of data is increased by adding noise to the original data or by GPT-2. Adding noise is efficient because it allows us to increase the amount of data without taking too much time, and it also allows us to increase the robustness of the data more efficiently. In contrast, in this study, GPT-2 increases the amount of training data so that it can flexibly respond to different sentence structures and paraphrases.

The principle on which this research is based is back-translation. Back-translation is a method used to extend a monolingual corpus into bilingual data and add these to the training data. Figure 1 shows the flow of back-translation. In the study by Edunov et al, 226M sentences were extended to bilingual data by a De–En translator [1]. This has become the state-of-the-art method in English–German translation, with a BLEU of 35.

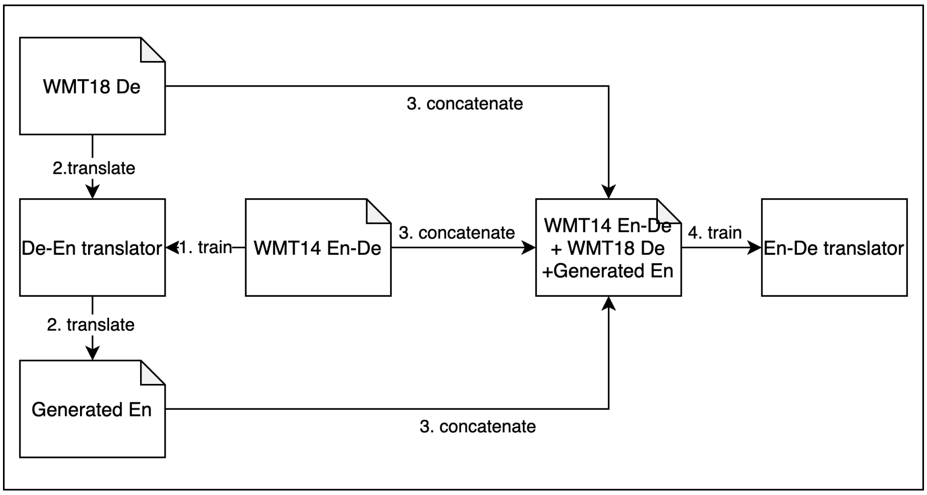

Figure 1. Back-translation is a method that uses additional data to improve translation performance. There are four major steps in the learning process. In recent years, it has begun to be used to evaluate state of the art models. This research is based on back-translation.

One of the major differences between back-translation and our work is in generating data and deciding whether to perform backward translation. First, back-translation uses a large dataset to learn the grammatical rules and knowledge contained in that dataset. In addition, adding source language side sentences to the dataset may help in generating more accurate sentences. In contrast, in this study, it takes time to generate the sentences to be added, and because the pretranslated language is added, the accuracy of the dataset depends on the performance of the original translation model. However, because there is no upper limit to the data expansion by the sentence generator, much more data can be added than in back-translation.

3. Neural Machine Translation

3.1. Transformer

The transformer is an attention-based architecture used to treat sequential flow like an RNN [8]. Unlike the RNN, transformers d not have recurrent structures and recognize word connections entirely by attention structures. Removing recurrent structures improves the computational inefficiency caused by the recurrent structure, which was the weak point of the RNN. Therefore, the RNN has been gradually replaced by the transformer, which has better recognition capability.

The transformer achieved a BLEU value of 28.4 in WMT14En-De in a 2017 experiment, indicating that it was the best translator at that time. Before that, ConvS2S Ensemble had achieved the highest BLEU score of 26.36.

Figure 2 shows the structure of the Transformer model. The transformer can be divided into two parts, an encoder and decoder, and the encoder part can be further divided into self-attention, feed-forward network and add and normalization parts [9]. To calculate self-attention, Equation (1) is used. Here, Q, K, and V are all matrices of size $\mathbb{R}^{T \times d_k}$, where T is the number of tokens and d_k is the size of the input vector.

$$Attention(Q,K,V) = softmax(QK^T/\sqrt{d_k})V \tag{1}$$

$$softmax(QK^T/\sqrt{d_k}) = \frac{exp(Q_iK_j^T/\sqrt{d_k})}{\Sigma_k exp(Q_iK_j^T/\sqrt{d_k})} \tag{2}$$

In addition, self-attention can learn multiple different patterns by using multi-head attention, which uses multiple query, key, and value matrices [10]. Multi-head attention is calculated according to Equation (3):

$$Attention_M(Q,K,V) = concat(H_1, H_2, H_M) \tag{3}$$

$$H_i = Attention(QW_Q^i, KW_K^i, KW_K^i) \tag{4}$$

where W_Q^i, W_K^i, KW_K^i are parameters to be learned and M is the number of heads. The output size of $Attention_M(Q,K,V)$ is the same as Q, K and V.

The feed-forward network is implemented as Equation (5).

$$FFN(z_i) = max(0, z_iW_1 + b_1)W_2 + b_2 \tag{5}$$

where z_i is the input vector and W_1W_2 are weight parameters and b_1b_2 are biases. W_1 and W_2 are $d \times d_h$ and $d_h \times d$. b_1 and b_2 are d_h and d dimension vectors.

Add and normalization has two functions: residual connection and layer normalization. Residual connection was proposed in an image classification paper [11]. Using residual connection, the network can preserve the previous state, and residual connection enables the training of multilayered networks. Layer normalization was proposed to apply a kind of batch normalization to RNN [12]. Layer normalization has also been used to good effect in transformers, and many multilayer transformers such as BERT and GPT-2 have some layer normalization in one layer. Layer normalization is implemented as Equations (6)–(9):

$$\hat{x}_i = \frac{x_i - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}} \tag{6}$$

$$\mu_L = \frac{1}{m}\Sigma_{i=1}^m x_i \tag{7}$$

$$\sigma_L^2 = \frac{1}{m}\Sigma_{i=1}^m (x_i - \mu_L)^2 \tag{8}$$

$$y_i = \gamma\hat{x}_i + \beta \tag{9}$$

where γ and β are learnable parameters and are initially set to 1 and 0.

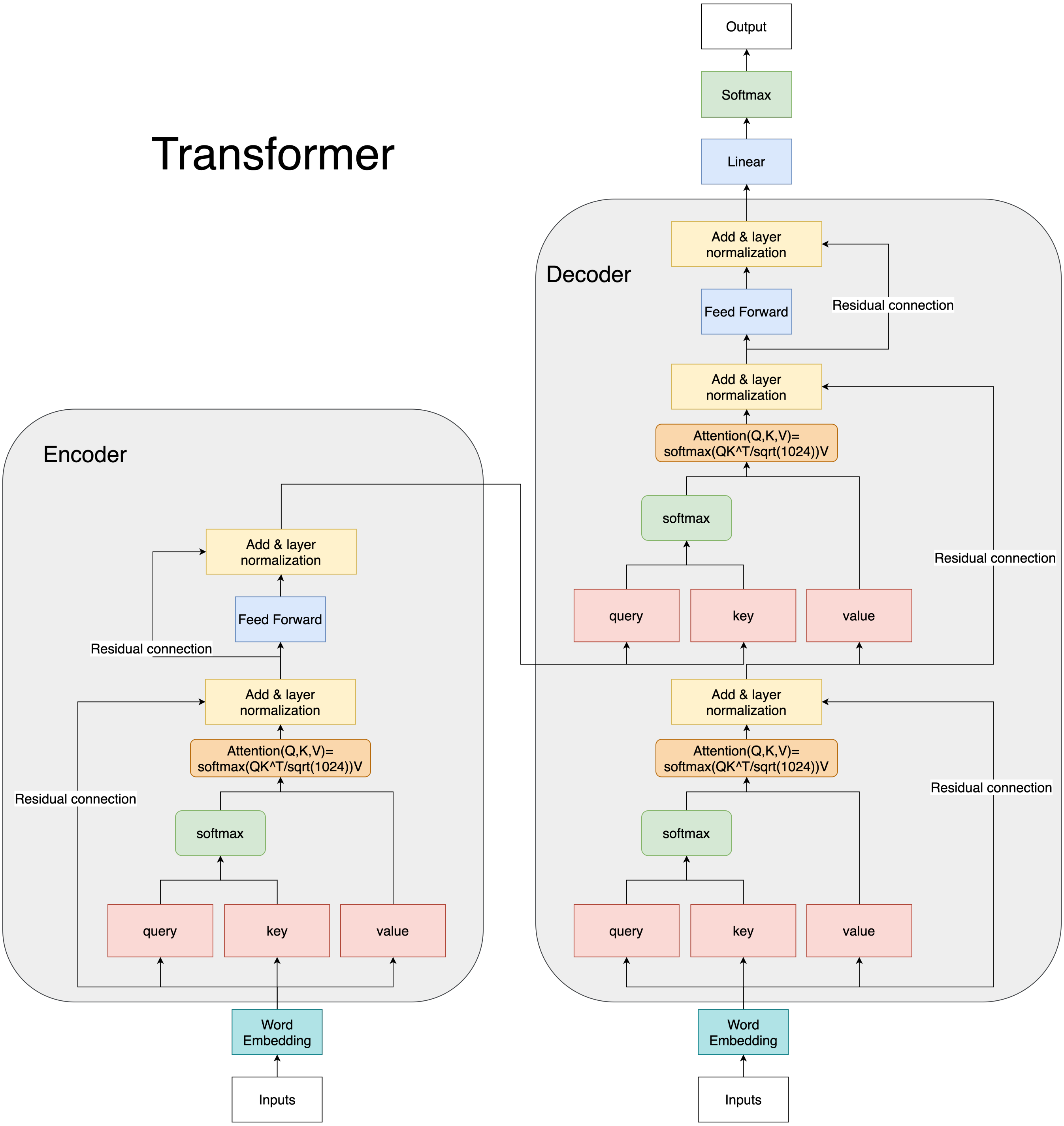

Figure 2. This is a diagram of the encoder–decoder of a transformer. The structure is the basis for BERT and GPT-2, which are used in this study.

3.2. BERT

Bidirectional Encoder Representations from Transformer (BERT) is a pretrained language representation model [13]. BERT is a pretraining model and is learned by a transformer processing the input unlabeled distributed representation. In practice, a transformer learns by simultaneously running two methods: the Masked Language Model and Next Sentence Prediction. The best aspect of BERT is that it can vectorize words to reflect the context [14]. This allows words with multiple meanings to be translated appropriately.

Figure 3 shows the structure of the BERT large model, which differs from transformers in three ways. First, we use Positional Embedding instead of Positional Encoding to learn better position vectors, and embedding is added to separate segments. Furthermore, BERT is used internally to increase the size of the vector once to obtain better representation.

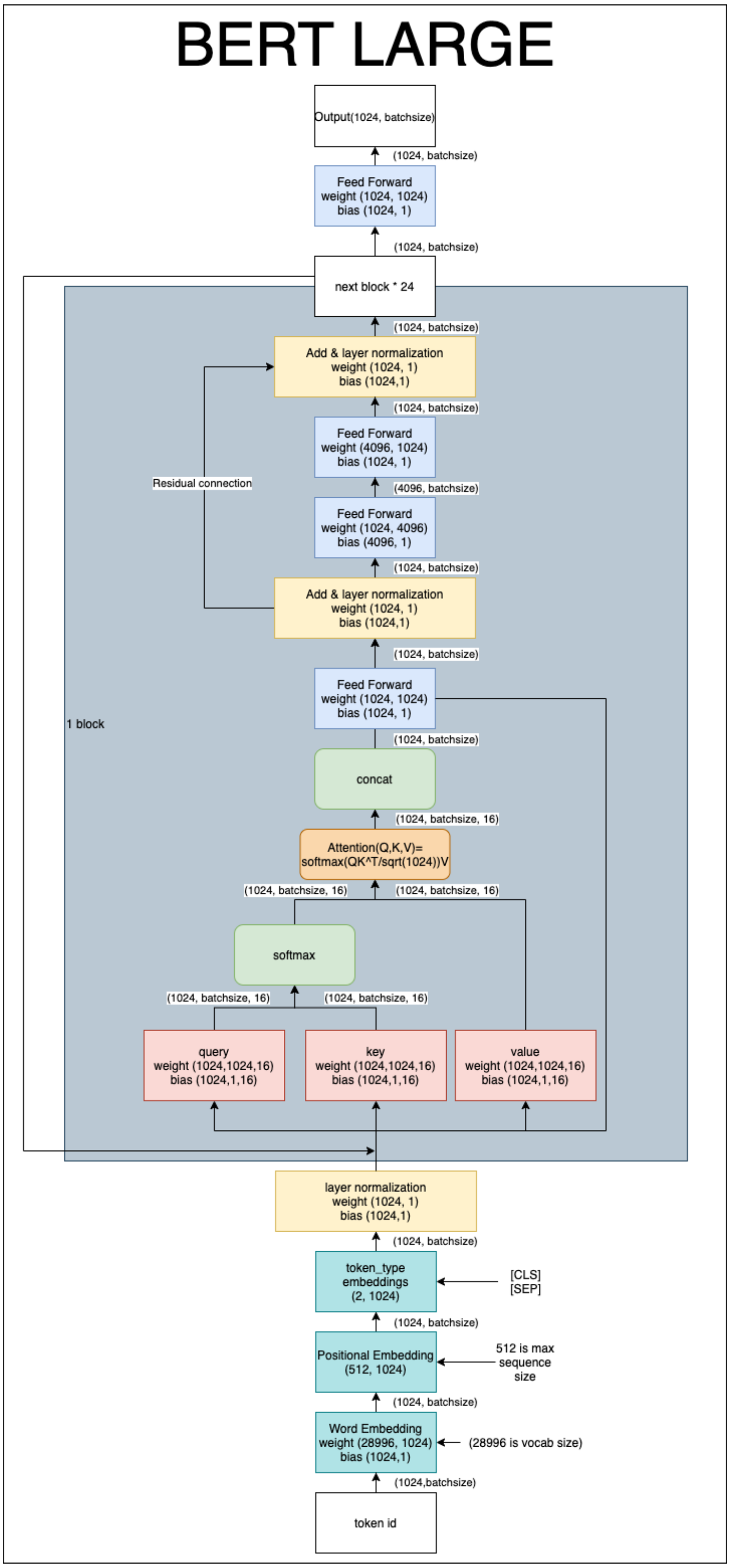
BERT LARGE
Output(1024, batchsize)
(1024, batchsize)
Feed Forward weight (1024, 1024) bias (1024, 1)
(1024, batchsize)
next block * 24
(1024, batchsize)
Add & layer normalization weight (1024, 1) bias (1024,1)
(1024, batchsize)
Feed Forward weight (4096, 1024) bias (1024, 1)
Residual connection
(4096, batchsize)
Feed Forward weight (1024, 4096) bias (4096, 1)
(1024, batchsize)
Add & layer normalization weight (1024, 1) bias (1024,1)
(1024, batchsize)
Feed Forward weight (1024, 1024) bias (1024, 1)
1 block
(1024, batchsize)
concat
(1024, batchsize, 16)
Attention(Q,K,V)= softmax(QK^T/sqrt(1024))V
(1024, batchsize, 16)
(1024, batchsize, 16)
softmax
(1024, batchsize, 16)
(1024, batchsize, 16)
query weight (1024,1024,16) bias (1024,1,16)
key weight (1024,1024,16) bias (1024,1,16)
value weight (1024,1024,16) bias (1024,1,16)
layer normalization weight (1024, 1) bias (1024,1)
(1024, batchsize)
token_type embeddings (2, 1024)
[CLS] [SEP]
(1024, batchsize)
Positional Embedding (512, 1024)
512 is max sequence size
(1024, batchsize)
Word Embedding weight (28996, 1024) bias (1024,1)
(28996 is vocab size)
(1024,batchsize)
token id

Figure 3. In the BERT Large model, the numbers next to the weight and bias indicate the size of the tensor, and the numbers with arrows indicate the size of the tensor flowing through it.

3.3. BERT-Fused

BERT-fused is a model that was proposed by Zhu et al. [15] in 2020 and is currently the best translation model that does not require additional data. Figure 4 shows how to use BERT's output. BERT-Enc Attention and BERT-Dec Attention select whether to use the BERT output or not. There have been many attempts to use BERT in translation, but this model improves performance by incorporating BERT calculations in parallel with regular transformer calculations. The BERT-fused model achieved a BLEU value of 30.75 in WMT14En-De. Furthermore, by also using back-translation, the BLEU score increased from 37.73 to 39.10 in the WMT16 Ro-En experiment. Therefore, we can say that it is also effective against back-translation.

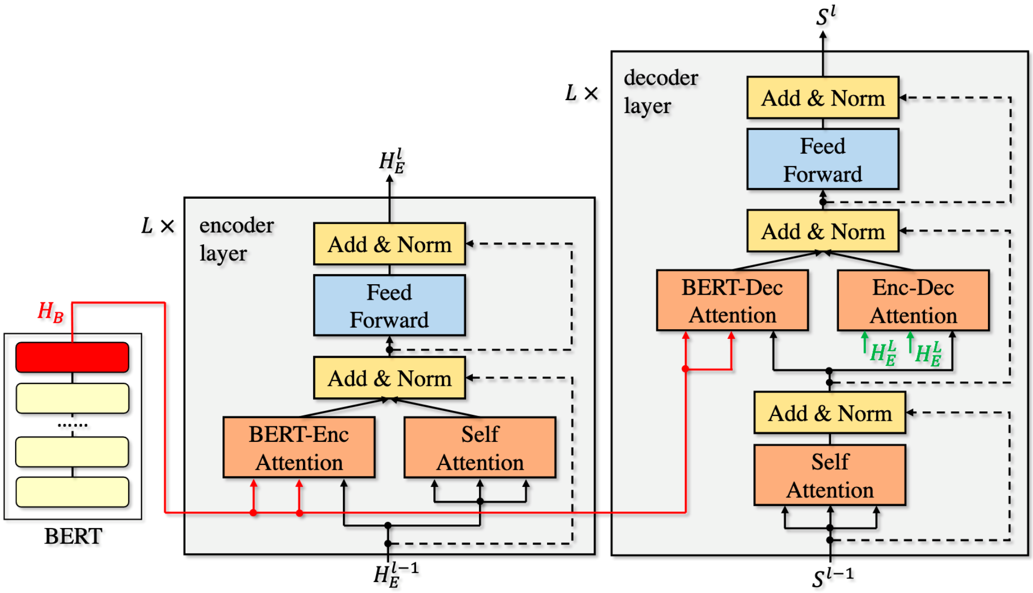

Figure 4. The BERT-fused model is a seq-seq model based on the transformer, and like the transformer, it has an encoder and a decoder. This is the model we use in the experiments of this study.

BERT-fused can be also divided into several parts. The main difference between the BERT-fused model and the regular transformer is the addition of BERT-Enc Attention and BERT-Dec Attention. The formula for attention with these added is as follows. h_i^l is the l-th layer's output of attention in the encoder part and s_t^l is the l-th layer's output of attention in the decoder part.

$$h_i^l = \frac{1}{2}(Attention(h_i^{l-1}, H_E^{l-1}, H_E^{l-1}) + Attention(h_i^{l-1}, H_B, H_B) \tag{10}$$

$$s_t^l = \frac{1}{2}(Attention(s_t^{l-1}, H_B, H_B) + Attention(s_i^l, H_E^L, H_E^L) \tag{11}$$

where H_B is the output of BERT and H_E^l is the hidden representation of the l-th layer in the encoder.

4. Sentence Generation

GPT-2

GPT-2 is a text generation model based on the transformer [16]. It can generate various types of text by using a dataset of 8 million web pages manually selected from links in sentences highly rated by users on the social bookmarking site Reddit. In particular, GPT-2 is suitable for generating continuation sentences, which are similar to human sentences.

Figure 5 shows the structure of GPT-2. GPT-2 is based on the decoder part of the transformer, and the output is a word. It also uses Positional Embedding, just like BERT.

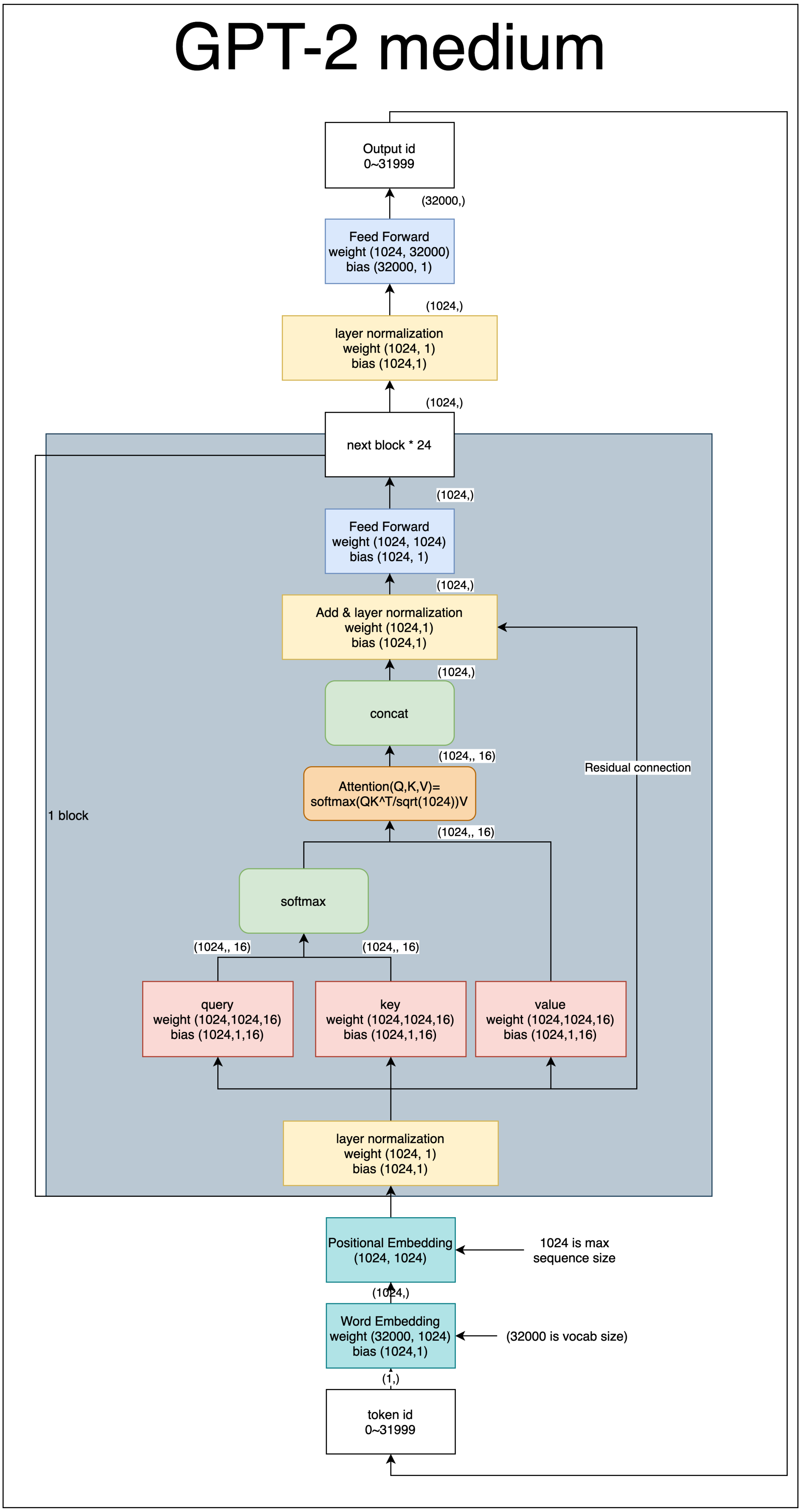

Figure 5. In the GPT-2 normal model, where the numbers next to the weight and bias indicate the size of the tensor, and the numbers with arrows indicate the size of the tensor flowing through it.

5. Experiment

5.1. Datasets

Our experiments used three kinds of data based on language distance because we considered that the performance of data augmentation for back-translation depends on the language distance with the same synthetic sources used for generation. As a short-distance language, an English–German (En–De) bilingual corpus was used; English–Japanese (En–Ja) was used as a long-distance language, and English–Chinese (En–Ch) was used as as medium-distance language. For En–Ja translation, the "Tatoeba" corpus was used. This corpus is rather small, but many texts are correctly translated [17]. For En–De translation, the WMT14 [18] corpus was used, and for En–Ch translation, we used the WMT18 corpus [19]. WMT datasets are large, but some sentence pairs are not translated correctly. We reduced the number of sentences in the WMT18En-Ch corpus to compare the score with the WMT14En-De experiment.

5.2. Data Generation and Processing

In this study, we used the text generation feature of GPT-2 to increase the sentence data. Unlike back-translation, to increase the data on the source side, we used the English version of GPT-2 published by OpenAI. Before generating the data, the data were split. The next step was to input the source language dataset into GPT-2 because the GPT-2 published by OpenAI is only available in English. Next, all the data were tokenized using mosesdecoder's tokenizer.perl. Next, we allowed the subword-nmt tool to learn the byte pair encoding (BPE) patterns of target data and apply the BPE to all data. Next, all data were cleaned using mosesdecoder's clean-corpus-n.perl. Using this cleaner, sentences longer than 250 words and sentence pairs with a source/target length ratio exceeding 1.5 were removed. Next, sentences were generated using GPT-2. To use GPT-2, first, the model must be downloaded using download_model.py, provided by OpenAI. Second, sentences were generated using interactive_conditional_samples.py. The top_k parameter used for Beam search was set to 40. After generating the data on the target side, the process from tokenizing to cleaning was applied to the data on the target side. Sentence generation was processed on one CPU (Intel(R) Xeon(R) W-2135 CPU @ 3.70GHz) and one GPU (Nvidia Quadro RTX 6000 24GB). This took 14 days for the tatoebaEn-Ja dataset, 43 days for the WMT14En-De dataset and 48 days for the WMT18En-Ch dataset. Table 1 is a summary of the amount of data. Table 2 is an example of GPT-2 generation.

Table 1. The amount of training, validation and test data, and the amount of data to be added to the training data by GPT-2 generation.

Dataset	tatoebaEn-Ja	WMT14En-De	WMT18En-Ch
Train Size	42,876	3,961,179	3,961,179
Valid Size	5404	40,058	40,058
Test Size	5404	3003	2002
Generated Data Size	1,673,293	4,657,103	5,437,102
After Removal	1,304,598	3,865,554	3,865,554

Table 2. An example of GPT-2 predicting the continuation of a sentence of tatoebaEn-Ja and WMT14En-De. In the case of the tatoeba corpus, a novel-like sentence is generated, while in the case of the WMT corpus, a news-like sentence is generated.

Original Sentence (tatoebaEn-Ja)
My mother set the table for dinner.
Generated by GPT-2
“I'm just going to start out with the turkey and then it'll get a bit more elaborate.” As the meal was being served, I saw his mother looking at me with a curious look of confusion. “I asked you to be the hostess?” The look made the girl seem confused by the idea. Her face was completely red. “Well… what kind of hostess do you think I am?” Her mother, who I guess was her wife, looked at me. Her eyebrows were raised. After hearing the words, her eyes widened and she opened them wide in shock. “What do you mean? Are you telling me you can be the hostess? I'm in a hurry. I'm going to be late, I'm going to be late!” And it was all over. My mother was standing at the table after the dinner and my aunt and uncle were looking at me with a curious expression. I was a bit amazed.
Original Sentence(WMT14En-De)
This is all in accordance with the principles that we have always upheld .
Generated by GPT-2
“We will continue to work to preserve the Constitution of the United States, and we will do so in accordance with the rule of law” and “to respect our fellow citizens no matter where they live or how many different languages we all speak, or what country they live in”. At a news conference today, the president also urged Congress to continue to work on “a balanced approach” that would also allow for the possibility of a bi-national test. “We have to look at what we would do with people coming in from other countries that are hostile to the U.S.”, he said. The New York City police department said it stopped arresting hundreds of people a day in 2012 because its officers didn't think their fellow officers were doing their jobs properly.

5.3. Model and Parameters

In this study, we used the BERT-fused model [15] without pretraining, and all learnable parameters were randomly initialized. The batch size was 4000 tokens, and we used Adam for optimization. The first learning rate was 0.0005, and the minimum learning rate was 1×10^{-9}. The reason we used the BERT-fused model is because it also works for back-translation. In this study, machine translation training was conducted in five steps. Figure 6 illustrates the flow of data generation and training. First, the base En dataset was inserted into GPT-2(1558M) to generate monolingual English data. Second, the En–Ja model was trained using the base dataset. Third, monolingual En data generated by GPT-2 were entered into the En–Ja translator. Fourth, the En–Ja model was trained by the generated dataset and the base dataset. Finally, the BLEU value was calculated and compared with other models.

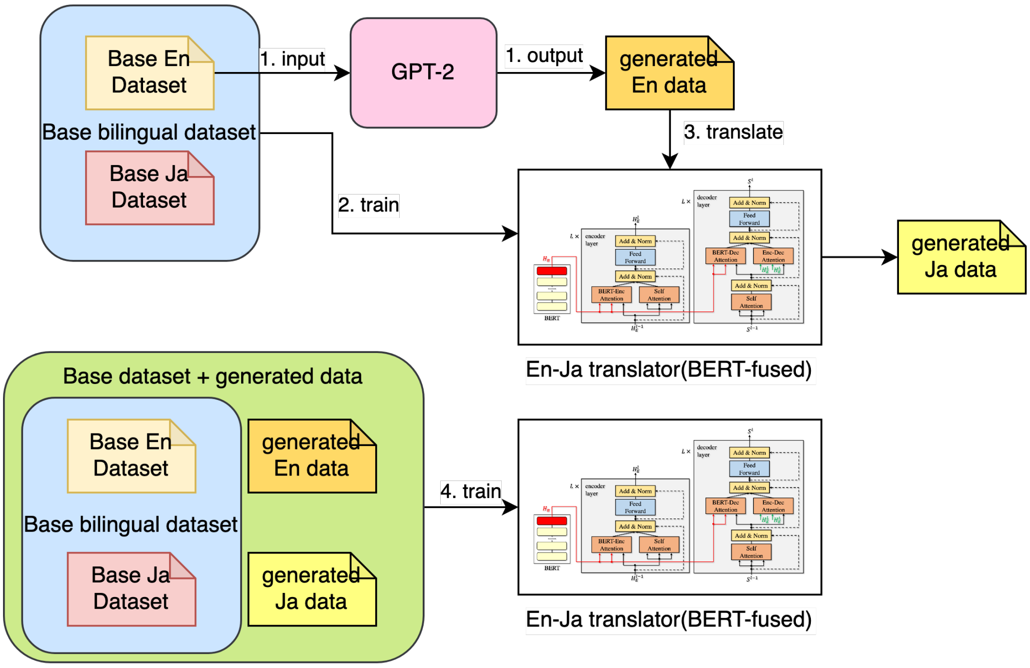

Figure 6. This figure describes how to increase the data and which data to use to train the translator.

6. Evaluation

We used the fairseq-score tool to calculate the BLEU value [20]. To compare how the BLEU value increased, this experiment used the BLEU-epoch line chart. For WMT14En-De, we also evaluated it by comparing it with the following four studies.

- Transformer BIG is an ordinary transformer; in a study using the WMT14 En-De, it had a BLEU score of 28.4.
- Back-Translation is back-translation with the Transformer BIG model. To compare with the current experiment, we adjusted the number of datasets to be added.
- Multi-Branch Attentive Transformer (MAT) is a translation model that uses a multi-branch architecture; it is based on the transformer model and has a BLEU score of 29.9.
- BERT-fused or BERT-fused (base) is an experiment without additional data using the BERT-fused model.

Figure 7 shows the result of the tatoebaEn-Ja experiments. We trained six different models in the experiment on the tatoebaEn-Ja dataset and compared the results. This study's BLEU value was the highest of the models compared in this experiment. Only 100,000 sentences were used for the back-translation, so the back-translation score is smaller than this study's score. Furthermore, the BERT-fused model's BLEU score increased from 25 to 27, meaning that the BLEU score increased with our proposed method.

Figure 8 shows the result of the WMT14En-De experiment, in which adding the data increased the BLEU value from 29.21 to 30.14. The BLEU value did not increase much compared with the tatoebaEn-Ja experiment, probably because the amount of data added to the original data was small.

Figure 9 shows the result of the WMT18En-Ch experiment, in which adding the data increased the BLEU value from 23.91 to 24.12. The results are similar to those of the WMT14En-De experiment.

Table 3 compares the scores between our experiment and results from other papers. In the WMT14En-De experiment, the BLEU value of back-translation was higher than that of the current model. This result indicates that the corpus added by back-translation contained more sentences suitable for translation than the sentences generated by GPT-2. Because the back-translation used in this study was based on German monolingual newscrawl data distributed with WMT18, more of the data were likely suitable for the WMT dataset [1].

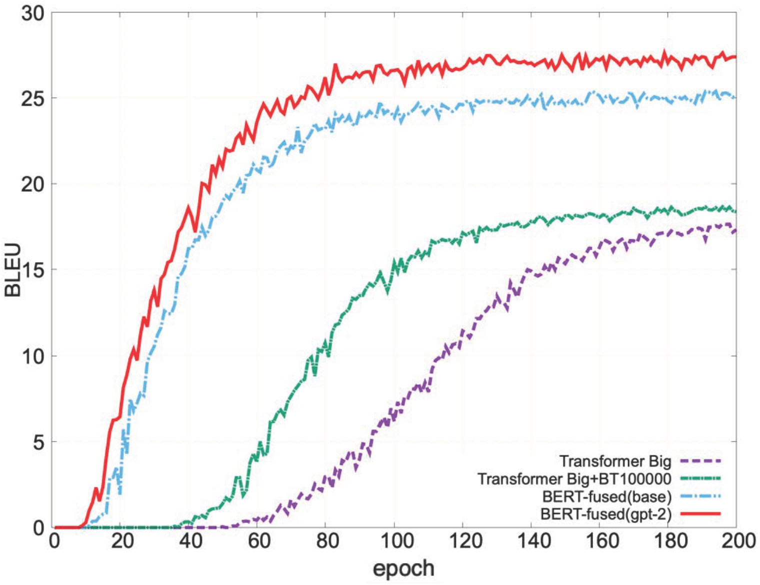

Figure 7. Results of the tatoebaEn-Ja experiment. tatoebaEn-Ja requires a larger number of epochs than the other experiments due to the small amount of data. Therefore, Figure 7 shows up to epoch 200.

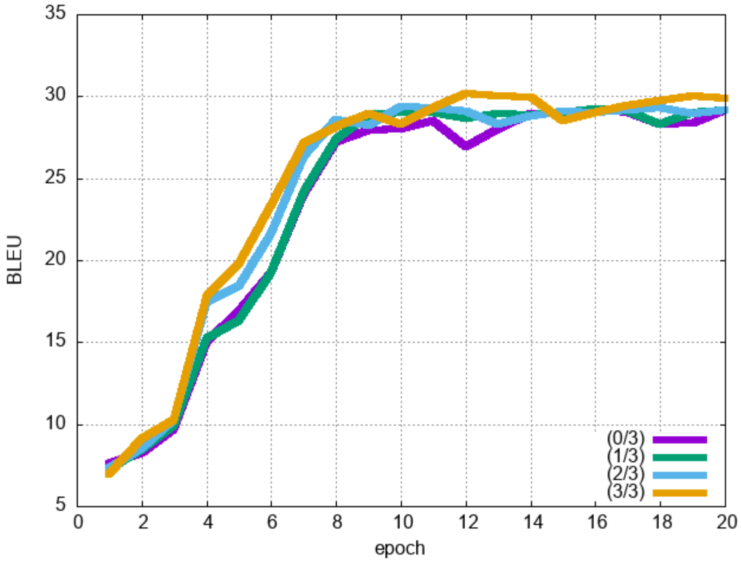

Figure 8. Results of the WMT14En-De experiment.

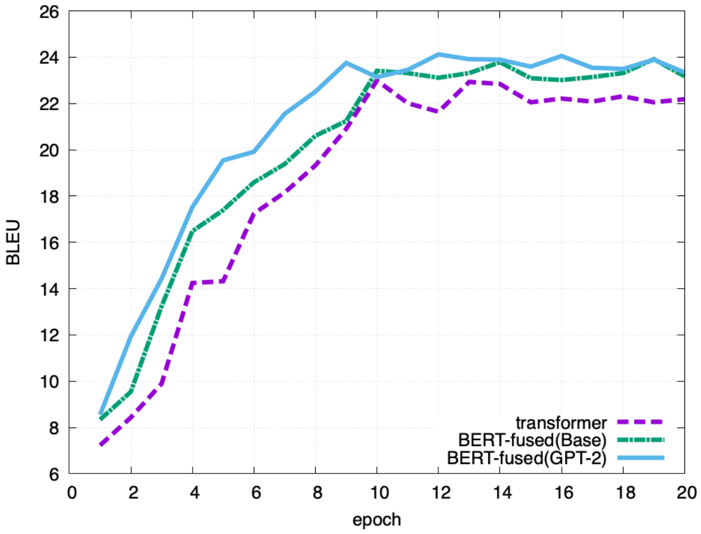

Figure 9. Results of the WMT18En-Ch experiment.

Table 3. A comparison table of the results of our experiment and those of other papers (shown with citations). Back-translation of WMT14En-De was conducted using the same amount of data generated by GPT-2. The experimental results examining tatoebaEn-Ja used 100,000 data points for back-translation. All other results are from experiments using the same parameters as in other papers.

Model	tatoebaEn-Ja	WMT14En-De	WMT18En-Ch
Transformer	19.67	28.4 [8]	22.99
Back-Translation	19.41	29.31	
MAT	22.96	29.9 [10]	
BERT-fused		30.75 [15]	
BERT-fused (base)	25.36	29.80	23.91
BERT-fused (GPT-2)	27.50	30.14	24.12

Next, we experimented with adjusting the amount of data to be added. For each dataset, we compared the data added by GPT-2 with one-third or two-thirds of the data used. Figure 10 shows the result of English–Japanese experiment, Figure 11 shows the result of English–German experiment and Figure 12 shows the result of English–Chinese experiment. In the English–Japanese experiment, the BLEU score increased with each additional amount of data. In the English–German experiment, the BLEU score increased with each additional amount of data. However, we did not see much difference between the version that used one-third of additional data and the version that did not. This may be because there was not much useful data in the third of the data added. Finally, in the English–Chinese experiment, as in the other experiments, the BLEU score increased with each additional amount of data.

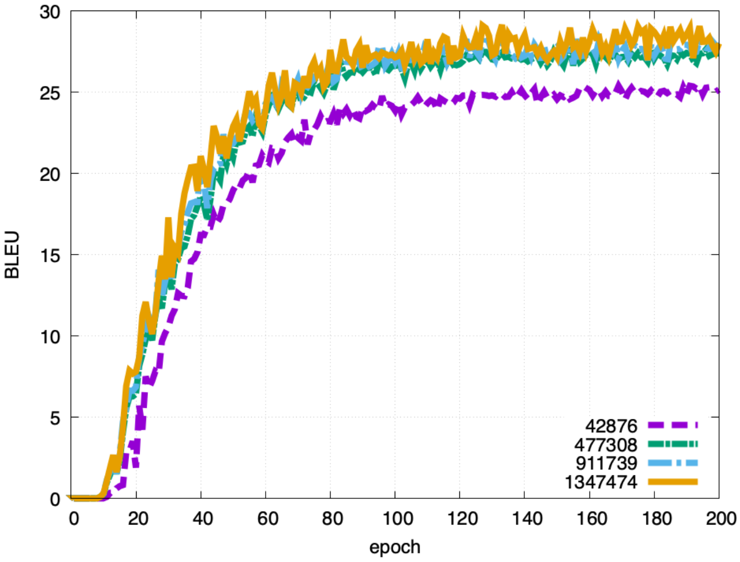

Figure 10. Effect of data size changes for English–Japanese translation. As in Figure 7, the results up to epoch 200 are shown.

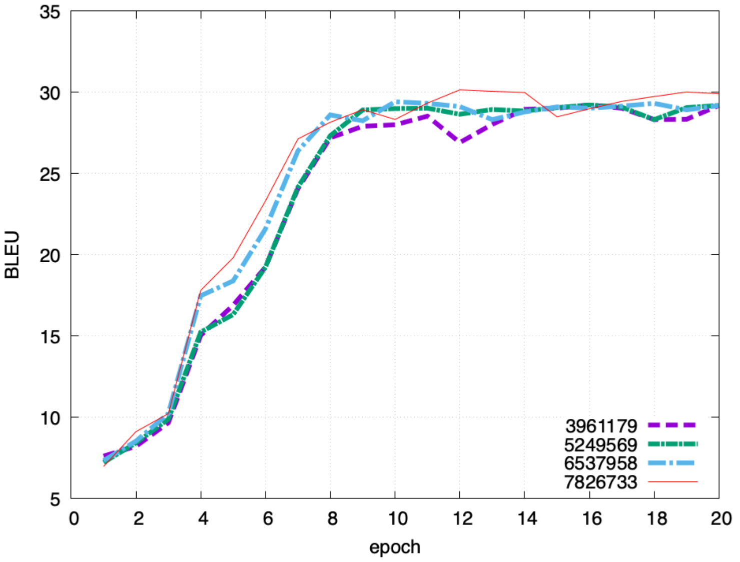

Figure 11. Effect of data size changes for English–German translation.

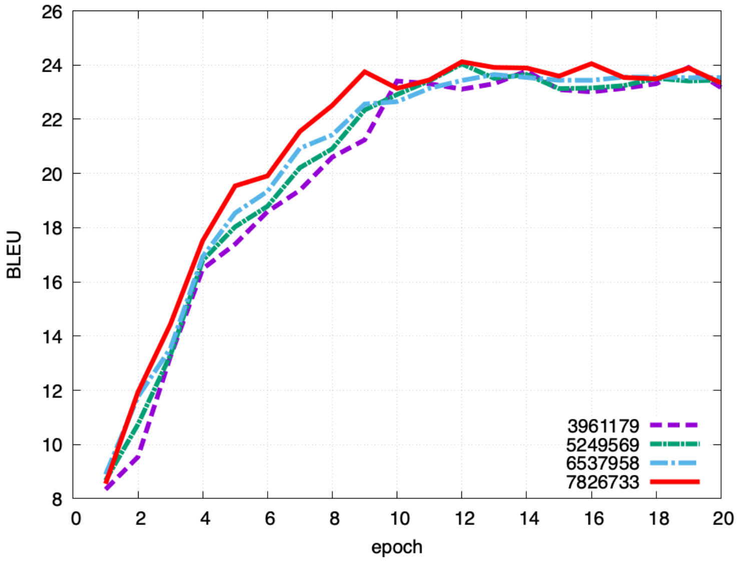

Figure 12. Effect of data size changes for English–Chinese translation.

7. Conclusions

In the cases of En–Ja and En–De, our method outperformed all selected methods. The experimental results on the tatoebaEn-Ja dataset were much higher than those on the WMT14En-De dataset, and we believe the reasons for this are as follows. First, this may be because of the relatively small amount of original data in the tatoebaEn-Ja dataset. Therefore, this method can be more beneficial when the amount of original data is small. This feature is the same as in the case of image processing [21]. Second, we found that the performance increases as the amount of data increases. The upper limit of this increase should be investigated further. In this experiment, it took a long time to generate sentences in GPT-2, and thus the preparation of sufficient data becomes an obstacle. Therefore, a data generation method in a pretrained model such as GPT-2 needs to be developed. In future work, we would be interested in examining the efficiency of the recent GPT-3 model [3] regarding sentence augmentation for language translation.

Author Contributions: Conceptualization, R.S.; methodology, R.S.; software, R.S.; validation, R.S.; formal analysis, R.S.; investigation, R.S.; resources, I.P.; data curation, R.S.; writing—original draft preparation, R.S.; writing—review and editing, R.S., I.P. and A.K.; visualization, R.S.; supervision, I.P.; project administration, R.S. and I.P.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank all the reviewers who contributed to the reviewing process. They are also thanful for the dedicated effort of the editorial board of MDPI Electronics journal.

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

References

1. Edunov, S.; Ott, M.; Auli, M.; Grangier, D. Understanding back-translation at scale. *arXiv* **2018**, arXiv:1808.09381.
2. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
3. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
4. 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, Beijing, China, 26–31 July 2015; Volume 1: Long Papers, pp. 1681–1691.

5. 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.
6. Duan, S.; Zhao, H.; Zhang, D.; Wang, R. Syntax-aware data augmentation for neural machine translation. *arXiv* **2020**, arXiv:2004.14200.
7. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
9. Fan, Z.; Gong, Y.; Liu, D.; Wei, Z.; Wang, S.; Jiao, J.; Duan, N.; Zhang, R.; Huang, X. Mask Attention Networks: Rethinking and Strengthen Transformer. *arXiv* **2021**, arXiv:2103.13597.
10. Fan, Y.; Xie, S.; Xia, Y.; Wu, L.; Qin, T.; Li, X.Y.; Liu, T.Y. Multi-branch Attentive Transformer. *arXiv* **2020**, arXiv:2006.10270.
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
12. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
13. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
14. Do, Q.M.; Paik, I. Resolving Lexical Ambiguity in English–Japanese Neural Machine Translation. In Proceedings of the 2020 3rd Artificial Intelligence and Cloud Computing Conference, Kyoto, Japan, 17–19 December 2020.
15. Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; Liu, T.Y. Incorporating bert into neural machine translation. *arXiv* **2020**, arXiv:2002.06823.
16. Radford, A. Better Language Models and Their Implications. *OpenAI Blog* **2020**, *1*, 2.
17. Tiedemann, J. The Tatoeba Translation Challenge–Realistic Data Sets for Low Resource and Multilingual MT. *arXiv* **2020**, arXiv:2010.06354.
18. Bojar, O.; Buck, C.; Federmann, C.; Haddow, B.; Koehn, P.; Leveling, J.; Monz, C.; Pecina, P.; Post, M.; Saint-Amand, H.; et al. Findings of the 2014 workshop on statistical machine translation. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, ML, USA, 26–27 June 2014; pp. 12–58.
19. Bojar, O.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Huck, M.; Koehn, P.; Monz, C. Findings of the 2018 Conference on Machine Translation (WMT18). In Proceedings of the Third Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2018; Association for Computational Linguistics: Belgium, Brussels, 2018; Volume 2: Shared Task Papers, pp. 272–307.
20. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the NAACL-HLT 2019: Demonstrations, Minneapolis, MN, USA, 2–7 June 2019.
21. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]