electronics

Article

A Scenario-Generic Neural Machine Translation Data Augmentation Method

Xiner Liu [1], Jianshu He [2], Mingzhe Liu [3,*], Zhengtong Yin [4], Lirong Yin [5] and Wenfeng Zheng [2,*]

[1] Division of Psychology and Language Sciences, University College London, Gower Street, London WC1E 6BT, UK
[2] School of Automation, University of Electronic Science and Technology of China, Chengdu 610054, China
[3] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou 325000, China
[4] College of Resource and Environment Engineering, Guizhou University, Guiyang 550025, China
[5] Department of Geography and Anthropology, Louisiana State University, Baton Rouge, LA 70803, USA
* Correspondence: liumz@cdut.edu.cn (M.L.); winfirms@uestc.edu.cn (W.Z.)

Abstract: Amid the rapid advancement of neural machine translation, the challenge of data sparsity has been a major obstacle. To address this issue, this study proposes a general data augmentation technique for various scenarios. It examines the predicament of parallel corpora diversity and high quality in both rich- and low-resource settings, and integrates the low-frequency word substitution method and reverse translation approach for complementary benefits. Additionally, this method improves the pseudo-parallel corpus generated by the reverse translation method by substituting low-frequency words and includes a grammar error correction module to reduce grammatical errors in low-resource scenarios. The experimental data are partitioned into rich- and low-resource scenarios at a 10:1 ratio. It verifies the necessity of grammatical error correction for pseudo-corpus in low-resource scenarios. Models and methods are chosen from the backbone network and related literature for comparative experiments. The experimental findings demonstrate that the data augmentation approach proposed in this study is suitable for both rich- and low-resource scenarios and is effective in enhancing the training corpus to improve the performance of translation tasks.

Keywords: neural machine translation; data augmentation; reverse translation; low-frequency word replacement; grammatical error correction

Citation: Liu, X.; He, J.; Liu, M.; Yin, Z.; Yin, L.; Zheng, W. A Scenario-Generic Neural Machine Translation Data Augmentation Method. *Electronics* **2023**, *12*, 2320. https://doi.org/10.3390/electronics12102320

Academic Editor: Fabio Grandi

Received: 4 April 2023
Revised: 15 May 2023
Accepted: 17 May 2023
Published: 21 May 2023

1. Introduction

In recent years, neural machine translation (NMT) systems have become the mainstream method in machine translation research and have achieved state-of-the-art results in various public translation tasks, replacing statistical machine translation [1–5]. Unlike statistical machine translation [6], neural machine translation uses an encoder–decoder framework that does not require artificial features or prior domain knowledge, allowing for automatic encoding and decoding through neural networks. Additionally, neural machine translation is a data-driven system that utilizes deep learning technology and relies on a model structure with many parameters. The performance of the translation system greatly depends on the quality of the parallel corpus used in low-resource domains, which is difficult and expensive to obtain, while a monolingual corpus is readily available for almost any language. Therefore, researchers have focused on using data augmentation methods based on limited bilingual corpus and incorporating monolingual corpus to enhance data and improve translation performance, which has been an important research direction in the field of neural machine translation for a considerable amount of time [7,8].

Neural machine translation technology originated from Bengio et al.'s neural network probabilistic language model in 2003, which represented discrete characters as continuous dense distributed vectors through neural networks, solving the sparse problem [9]. In the sequence-to-sequence model, the input is encoded by the encoder into a fixed-length

context representation vector, and the decoder decodes this vector to obtain the output. However, fixed-length vectors have a better representation for short sentences but often cannot effectively represent long sentences. In 2014, Bahdanau of Youngor University in Germany proposed an attention mechanism that effectively solved this problem and brought machine translation to a new level [10]. The attention mechanism is essentially a small neural network that is trained at the same time as the "RNN-RNN" network, making the "RNN-RNN" model discriminative, so it can focus on more relevant input information. Luong et al. of Stanford proposed many variants of the attention mechanism to further enhance its representational ability [11].

Subsequently, research on neural machine translation mainly focused on the deformation of the encoder–decoder structure and the improvement of the attention mechanism. Bahdanau et al. enhanced the ability of the encoder to represent information by using a bidirectional recurrent neural network (Bi-RNN) for encoding [10]. Some scholars later studied decoder deformation. For instance, Liu et al. researched decoding from two directions respectively and finally combined the decoded content of the two directions [12], while Zhou et al. proposed a bidirectional simultaneous decoding method that dynamically determines the decoding direction of each word [13]. In 2015, Baidu applied neural machine translation technology to the online automatic translation platform and proposed a multi-channel encoder model that largely solved the problem of a lack of source information in the form of fusion encoding [14]. The team of Shen Shiqi from Tsinghua University proposed the minimum risk training criterion to deal with the mismatch between training and testing in the attention mechanism encoding and decoding framework, significantly improving the performance of machine translation [15]. Tu Zhaopeng of Huawei's Noah's Ark team proposed several techniques, such as reconstructor, mechanism, and historical and future information modeling, which significantly improved the translation quality and alignment accuracy and solved the problem of actual translation to some extent [16–18]. Nguyen et al. proposed a data-diversification-based method to improve NMT performance, which uses the predictions of multiple forward and backward models to diversify the training data [19]. Additionally, Xie, S. et al. proposed an end-to-end algorithm to handle entity translation, where the encoder and the decoder are attached to an entity classifier to treat named entities differently and improve the translation quality [20].

Data augmentation techniques have been extensively used and found effective in computer vision [21]. In machine translation, the most popular method of data augmentation is currently back translation. This method employs a target-to-source translation model, also known as a back translation model, to generate a pseudo-bilingual corpus, which is then used to train the source-to-target translation model, or the forward translation model [22,23]. Researchers have explored the use of back translation for context-aware neural machine translation and evaluated its impact on translation accuracy [24]. However, in low-resource settings, researchers have discovered that high-quality pseudo-parallel corpora can benefit the model due to the limited availability of bilingual data. In contrast, in rich-resource settings, beam search can produce high-quality translations that often focus on common words. As a result, the generated pseudo-parallel corpus may lack diversity, which hinders its ability to accurately represent the actual data distribution, and consequently, the performance improvement of the forward model may be limited. Therefore, current back-translation methods face the challenge of balancing the demands of quality and diversity in both low-resource and rich-resource scenarios [25].

This study analyzes the characteristics and limitations of various mainstream data augmentation methods and proposes a scene-agnostic neural machine translation data augmentation method. The method combines low-frequency word replacement and back-translation to further enhance the training corpus. In the experiment section, the WMT2015 English–German dataset is used, and rich-resource and low-resource scenarios are divided for comparative experiments from two perspectives. Firstly, a comparative experiment is conducted on different backbone networks. Secondly, a comparative experiment is conducted on related data augmentation works. The results in both rich-resource and

low-resource scenarios demonstrate that the proposed method can effectively enhance data and outperforms related works. This study solves the problem that mainstream data augmentation methods cannot be applied to rich-resource scenarios and low-resource scenarios at the same time.

2. Dataset and Methods

2.1. Dataset Preparation

To compare with related research [25,26], the rich-resource scenario uses the parallel bilingual corpus in the WMT2015En-De English–German translation task as the training data set, merges the official test data of newstest2014 and newstest2015 as the validation set, and adopts newstest2016 as the test set. The WMT2015En-De English–German dataset contains a total of 4.6 M parallel corpora of sentence pairs, which belong to the news field corpus.

In the experiment, the preprocessing steps include:

(1) Filter the sentences whose sequence length exceeds 50 in the original corpus.

(2) Filter the sentences whose length ratio of source language (English)/target language (German) exceeds 1.5.

(3) Use the Moses tool to tokenize the source language and the target language respectively. Moses' tokenizer splits the text into individual words and adds a space after each word, with the exception of the final word in the sentence, which is followed by a period.

(4) Use the BPE tool to process the English–German bilingual corpus, train the shared subword dictionary [27], and set the size of the subword dictionary to 40 k;

(5) Use the trained BPE tool to segment the corpus, and add <bos> and <eos> identifiers at the beginning and end.

The results before and after data preprocessing are shown in Figure 1.

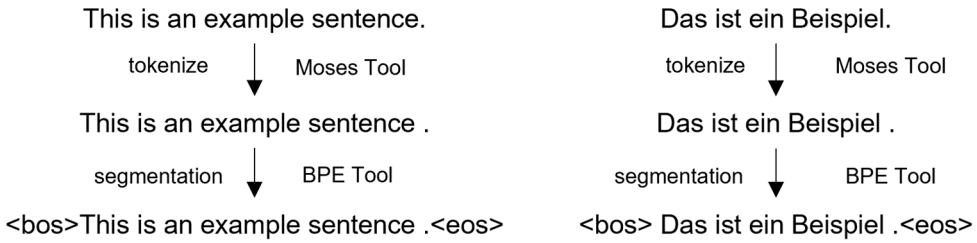

Figure 1. Data Preprocessing Results.

After preprocessing, the training corpus has a total of 3,904,999 sentence pairs.

The News Crawl Corpus crawled from the Internet is pre-cleaned first. The steps include: (1) remove html tags; (2) remove third-party language sentences; (3) remove sentences with a length of more than 50; after screening, the final monolingual corpus size is 2,002,899 sentences. In summary, the information on the datasets used in the rich-resource scenario experiment is shown in Table 1.

Table 1. Dataset Information of Rich-resource Scenarios.

Type		Name	Size
parallel corpus	Training set	WMT2015En-De	3.9 M
	validation set	newstest2014 + newstest2015	6 k
	test set	newstest2016	3 k
monolingual corpus		News Crawl Corpus	2 M

In the low-resource scenario, the same as the work of Edunov et al. [25], select 10% of the WMT2015En-De English–German dataset to simulate the low-resource scenario. The selection of the verification set and test set is the same as that of the rich-resource scenario,

and the monolingual corpus is also selected from the official News Crawl Corpus. The specific information of the dataset is shown in Table 2.

Table 2. Dataset information of low-resource scenarios.

Type		Name	Size
parallel corpus	Training set	WMT2015En-De	400 k
	validation set	newstest2014 + newstest2015	6 k
	test set	newstest2016	3 k
monolingual corpus		News Crawl Corpus	2 M

2.2. Auxiliary Module Preparation

(1) Language model preparation. The language model is consistent with the work of Fadaee et al. [26], choosing a bidirectional gated unit model (Bi-GRU), which is pre-trained in advance using a monolingual corpus.

(2) Align model preparation. The low-frequency word replacement module needs to obtain the position of the word to be replaced in the target language through the alignment model, so it needs to use the existing parallel corpus to pre-train the alignment model. Adopt fast_align [28] as the alignment model.

(3) Syntax error correction module preparation. The grammatical error correction method adopts the open-source method of Zhao et al. [29], which uses the method of adding noise to generate many grammatically incorrect sentences, and combines them with real sentences to form training data to train a grammatical error correction module. This article uses the English and German grammar error correction (GEC) model pre-trained and open sourced by Zhao et al.

(4) Translation quality evaluation. The translation results generated by the model in the test set are evaluated by the BLEU indicator [30], and the BLEU value is calculated using the multi_bleu.perl [31] script provided by the Moses tool.

2.3. Back Translation

Figure 2 demonstrates the specific flow of the reverse translation method [32]. It is assumed that in order to obtain a forward translation system $M_{x\to y}$, there are parallel bilingual corpus $D = \left\{x^{(n)}, y^{(n)}\right\}_{n=1}^{N}$ and target language monolingual corpus $Y = \left\{y^{(t)}\right\}_{t=1}^{T}$.The first step is to train the reverse translation system using bilingual corpus $D = \left\{x^{(n)}, y^{(n)}\right\}_{n=1}^{N}$, that is $M_{y\to x}$, the translation model from the target language to the source language. The second step is to decode the monolingual corpus $Y = \left\{y^{(t)}\right\}_{t=1}^{T}$ to obtain the translation results $x^{(t)}$, so as to construct the pseudo bilingual corpus $\widetilde{D} = \{x^{(t)}, y^{(t)}\}_{t=1}^{T}$. The third step is to combine the real bilingual corpus $D = \left\{x^{(n)}, y^{(n)}\right\}_{n=1}^{N}$ and pseudo bilingual corpus $\widetilde{D} = \{x^{(t)}, y^{(t)}\}_{t=1}^{T}$, and all the combined data are used to train the forward translation system.

In resource scenarios, adding some noise to the source language sentences generated by back translation to improve their diversity can achieve better results. In machine translation, beam search is most commonly used for decoding, but in a rich-resource context, beam search will cause translation results to mainly focus on some high-frequency words, and the generated pseudo data lack diversity [33]. To solve the problem of insufficient diversity of pseudo-corpora in rich-resource scenes, Edunov et al. used the methods of sample decoding, top-k decoding, and beam-plus-noise in their research. Sampling decoding in the process of generating translation results, all words in the vocabulary are taken into account and randomly sampled according to their predicted probability, so some words with low predicted probability may also be selected, which makes the translation results more diverse, but the translation quality and fluency will be greatly compromised. Top-k decoding combines beam search and sampling decoding. First, the first k words with

the highest probability in the predicted probability are selected, and random selection is performed among these k words, which can not only ensure the quality of translation but also have a certain diversity. The beam-plus-noise approach is to add noise to the translation results of beam search, such as dropped words, masked words, and random exchange of words.

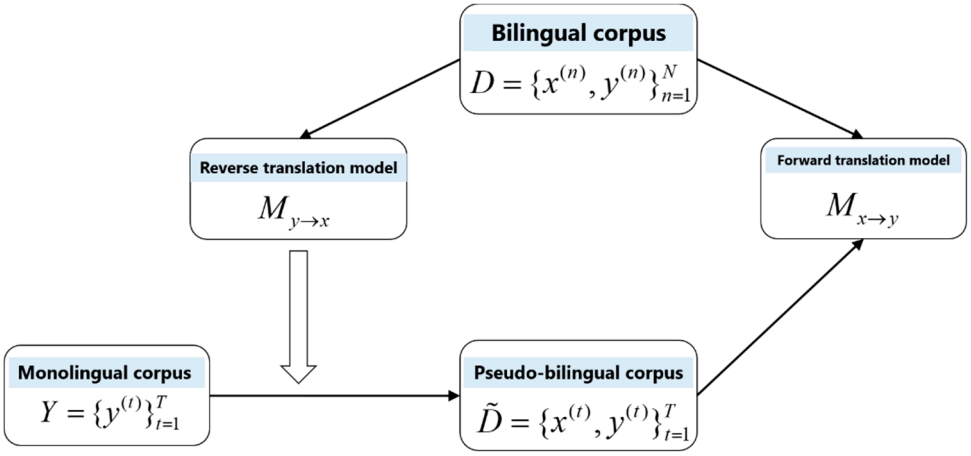

Figure 2. Reverse translation method.

2.4. Low-Frequency Word Replacement

In addition to the back-translation method, another method of data augmentation is adding noise. Common methods of adding noise include dropping words, masking words, and shuffling the order. The method of adding noise is generally only carried out on the source language sentences. The reason is to maintain the fluency of the target language sentences and ensure that the decoder is fully trained. On this basis, adding noise to the source language can increase the diversity of data to improve the robustness and generalization ability of the encoder [34]. Word replacement, as the name suggests, is to replace some words in a bilingual corpus with other words in the vocabulary, which is a unique method of adding noise. This method changes the semantics by substituting words but is generally syntactically sound. For example, for the sentence "Biden/won/election", you could replace "Biden" with "Trump", "Obama", or "won" with "lose", or "Election" with "match", "victory", etc.

Low-frequency word replacement replaces a word in the source language sentence with a rare word that satisfies the grammatical and semantic conditions through the language model, and then uses the alignment tool to find the corresponding position of the replaced word in the target language sentence, and uses the translation dictionary to convert the target language. The word at this position is also replaced with the corresponding translation result, thereby obtaining a pseudo-bilingual corpus. Rare words can easily lead to insufficient training due to their small number of occurrences, and such methods can greatly alleviate this situation.

2.5. Grammar Error Correction

The function of the grammatical error correction module is to detect whether there are grammatical errors in a sentence and automatically correct the detected grammatical errors. Figure 3 is an example of a grammatical error correction task. In this sentence, the original word absolute is grammatically incorrect, and the GEC module should absolutely recognize and modify it as an adverb.

Nothing is [absolute ⟶ absolutely] right.

Figure 3. Example of syntax error correction.

Currently, the task of correcting grammatical errors uses an encoder–decoder framework similar to neural machine translation. This framework can be understood as taking grammatically incorrect sentences as an input in the source language and outputting grammatically correct sentences in the target language. Figure 3 shows a pair of parallel training data, "Nothing is absolute right" and "Nothing is absolutely right". When a large-scale parallel corpus, such as <wrong sentence, correct sentence>, is used to train a GEC model, the model can automatically correct grammatical errors.

To enhance the performance of the GEC module, some researchers propose constructing pseudo-training data. Zhao et al. [29] constructed pseudo-training data by randomly adding noise, which achieved good performance. The method of adding noise involves randomly deleting a word with a 10% probability, randomly adding a word with a 10% probability, randomly replacing a word with a 10% probability, and increasing the serial number of all words to a normal distribution and reorder, resulting in an error sentence. Tao Ge et al. [35] used the reverse translation method of neural machine translation to train a wrong sentence generation model, which was then used to construct pseudo-training data and improve the performance of the GEC module. Alokla, A. et al. [36] proposed a new retrieval-based transformer pseudocode generation model that can handle low-frequency words and words that do not exist in the training dataset, which can be used in grammatical error correction tasks. In this study, the pre-trained GEC module of Zhao et al. is used.

2.6. Scenario-Generic Neural Machine Translation Data Augmentation Method

The general neural machine translation data enhancement method proposed in this paper, based on reverse translation, uses the word replacement method to replace common words with low-frequency words and reduces grammatical errors through the grammatical error correction module; the final generated corpus is parallel to the original. The corpus is merged into a bilingual corpus with both quality and diversity. As shown in Figure 4, it is a flow chart of this method.

The specific steps are:

(1) The existing limited parallel bilingual corpus $D = \left\{x^{(n)}, y^{(n)}\right\}_{n=1}^{N}$ and a large amount of target language monolingual corpus $Y = \left\{y^{(t)}\right\}_{t=1}^{T}$ first use the bilingual corpus to train the back translation model $M_{y \to x}$ from the target language to the source language;

(2) Determine the low-resource or rich-resource scenario, input the target language monolingual corpus into the $M_{y \to x}$ model, select the cluster search method to generate the translation $\widetilde{X} = \{x^{(t)}\}_{t=1}^{T}$ in the low-resource scenario, and select the Top-k decoding method to generate the translation in the rich-resource scenario. Use the Back Translation method of data augmentation corpus, $D_{BT} = \left\{x^{(t)}, y^{(t)}\right\}_{t=1}^{T}$;

(3) For the enhanced bilingual corpus after combining the translation generated by the reverse translation part with the original bilingual corpus, input the low-frequency word replacement module;

(4) The corpus is further enhanced by the low-frequency word replacement module, and the specific steps are shown in Figure 5. First, after artificially generating the low-frequency word set V, select a word s_i of the source language sentence S in the original sentence pair as the word to be replaced, and the language model will extract words from the low-frequency word set V as the replacement word for s_i. At this time, through the language model, calculate the probability distribution of the replaced sentence, and select the word with the largest probability distribution as the replacement word s'_i, as shown in Formula (1). Formula (1) shows that when s_i is replaced by s'_i in the low-frequency word set V, the language model will calculate the probability distribution before and after the replacement word, respectively, obtain certain candidate words, and the final replacement word s'_i takes the intersection. Then, through the alignment model, it is known that the corresponding word of s_i in the target language sentence T is t_j, and t_j is replaced with the

translation word t'_j of s'_i. Finally, the enhanced sentence pair $D_{WS} = \left\{x^{(k)}, y^{(k)}\right\}_{k=1}^{K}$ after the replacement of low-frequency words is obtained.

$$\begin{cases} \overrightarrow{C} = \{s'_i \in V : P_{forwardLM}(s'_i|s_1, \ldots, s_{i-1})\} \\ \overleftarrow{C} = \{s'_i \in V : P_{backwardLM}(s'_i|s_n, \ldots, s_{i+1})\} \\ C = \{s'_i|s'_i \in \overrightarrow{C} \wedge s'_i \in \overleftarrow{C}\} \end{cases} \tag{1}$$

(5) When in a low-resource scenario, the corpus generated after the low-frequency word replacement processing will also go through the grammar error correction module as post-processing to reduce grammatical errors in the corpus. The flowchart is shown in Figure 5.

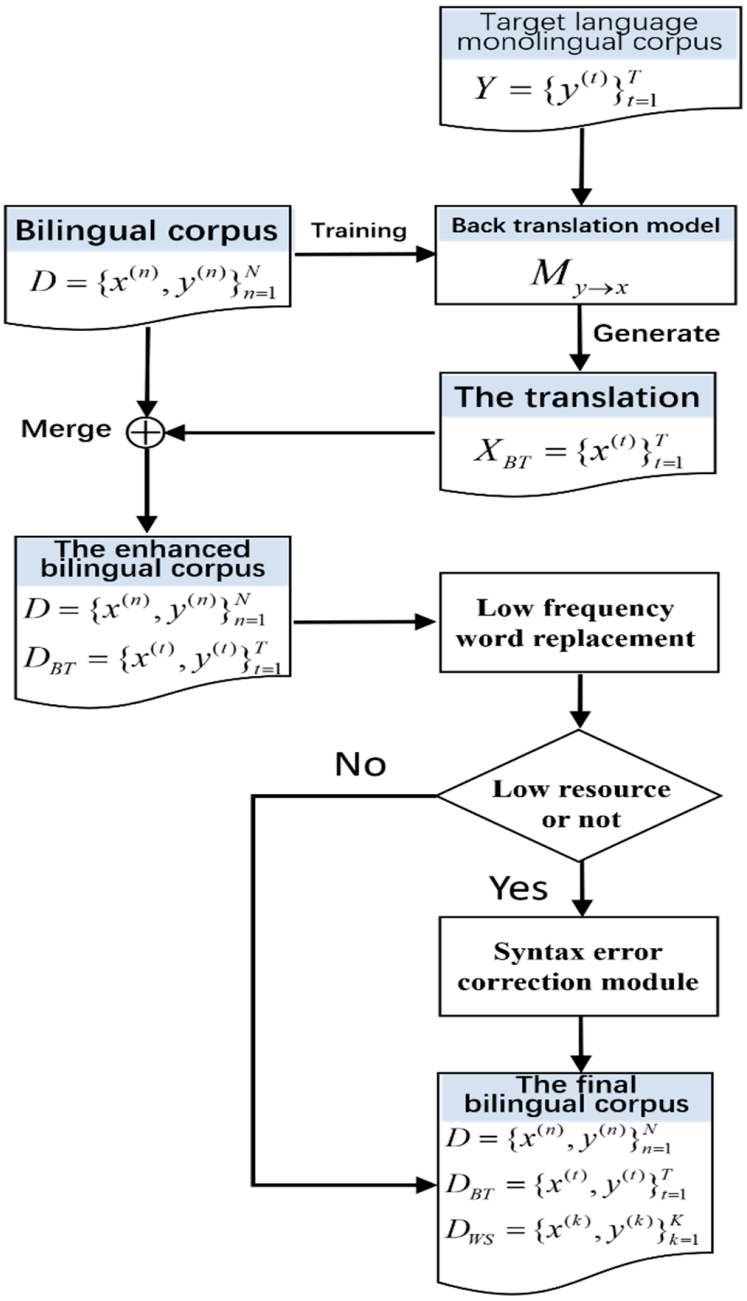

Figure 4. Flowchart of data enhancement methods common to scenarios.

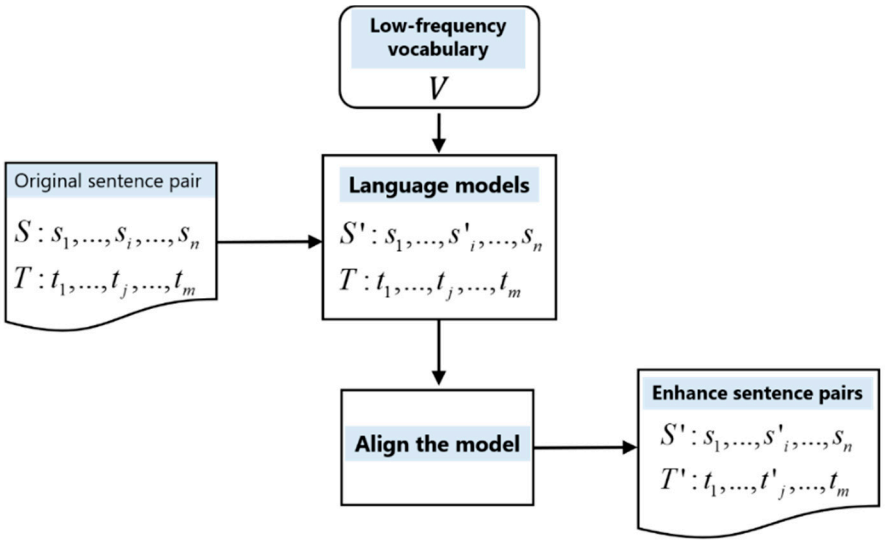

Figure 5. Schematic diagram of low-frequency word replacement module.

3. Experiment and Results

3.1. Baseline Models and Comparison Methods

The data enhancement method proposed in this paper combines low-frequency word replacement and reverse translation methods. The experiments are carried out from two perspectives to verify the effect:

To compare different backbone networks in neural machine translation, this study sets up experimental baseline models in three categories based on different backbone networks.

(1) RNNSearch: The work of Bahdanau et al. [10] in 2015, a milestone in the field of neural machine translation. Through Bi-LSTM combined with an attention mechanism, neural machine translation surpasses statistical machine translation;

(2) ConvS2S: The work of Gehring et al. [37] of the Facebook team in 2017 greatly improved the computational efficiency through convolutional neural networks and also improved the translation performance;

(3) Transformer-base: The work of Vaswani et al. of the Google team in 2017 presented a completely self-attention-based structure that achieved state-of-art effects on a large range of data sets and has now become the mainstream backbone in the field of neural machine translation networks.

Table 3 summarizes the advantages and disadvantages of RNNSearch, ConvS2S, and Transformer-base models for neural machine translation tasks:

Table 3. Advantages and disadvantages of backbone networks.

Model	Advantages	Disadvantages
RNNSearch	Can model sequential dependencies well Good for long sequences Easy to implement	Prone to vanishing gradients Computationally expensive
ConvS2S	Less prone to vanishing gradients Can capture long-term dependencies	Requires large datasets to train effectively Needs careful hyperparameter tuning
Transformer-base	Captures global dependencies better than RNNs Better for long sequences Faster than RNNs	Needs large amounts of data to train effectively More complex to implement than RNNs

Second, this study essentially proposes a data augmentation method, so it is compared with related work on data augmentation, as follows:

(1) base: only the original parallel corpus is used, and no data augmentation method is used.

(2) Edunov: The work of Edunov et al. [25] of the Facebook research team in EMNLP2018 uses the back translation method and verifies the effect of the back translation method for low-resource and rich-resource scenarios. It achieved first place in English–German translation in WMT2018.

(3) Fadaee: The work of Fadaee et al. [26] in ACL2017, the work carried out a word replacement method for low-frequency words, and selectively enhanced the training corpus.

(4) Our method; the data augmentation method proposed in this chapter combines back translation and low-frequency word replacement.

(5) Our method (+GEC): The method proposed in this chapter. Different from our method, in low-resource scenarios, additional syntax error correction (GEC) processing is performed on the generated pseudo-parallel corpus.

3.2. Environmental Preparation

This experiment is based on the deep learning framework of the system, and the specific environment settings are shown in Table 4.

Table 4. Environment Settings.

Hardware Configuration		**System Configuration**	
cpu	Intel i7-9700k	operating system	Ubuntu18.04LTS
Memory	32 GB	Development language	Python 3.6
graphics card	Nvidia RTX 2080ti	frame	Pytorch 1.5.0

3.3. Baseline Model Parameter Settings

(1) RNNSearch parameter settings

In the RNNSearch model, the number of hidden layer neurons is set to 1000, the word embedding dimension is set to 620, the number of attention layer neurons is set to 1000, the Adadelta optimizer is selected, the initial learning rate is set to 5×10^{-4}, and the batch size is set to 128.

(2) ConvS2S parameter settings

In the ConvS2S model, the number of hidden neurons of the encoder and decoder is set to 512 dimensions, the word embedding dimension is 512, the initial learning rate is 0.25, the batch size is set to 64, the dropout probability is set to 0.2, and the label smoothing is set to 0.1.

(3) Transformer-base model parameter settings

The number of encoder and decoder layers in the Transformer-base model is 6 layers, the number of heads of the multi-head attention mechanism is 8, the model dimension d_model is set to 512 dimensions, the feedforward neural network dimension d_ff is set to 1024 dimensions, and the $dropout$ probability is set to 0.1. The beam search width $beam\_width$ is set to 4 and label smoothing is set to 0.1.

The activation function during model training uses ReLU, and the optimizer uses Adam [38]. The Adam optimizer will dynamically adjust the learning rate during the training process. First set $warm\_step$; within $warm\_step$, the learning rate will increase linearly. After $warm\_step$, the learning rate will gradually decay, and the relevant parameter settings of the Adam optimizer are shown in Table 5.

Table 5. Adam optimizer parameter settings.

Parameter Name	**Parameter Meaning**	**Parameter Value**
β_1	First Moment Estimation Exponential Decay Rate	0.9
β_2	Second Moment Estimation Exponential Decay Rate	0.98
$warm\_step$	Model start steps	4000

3.4. Experimental Results

In this paper, experiments are carried out in the rich-resource scenario and the low-resource scenario, respectively, and the experiments performed in different scenarios are the same. As mentioned above, the baseline model is based on three backbone networks, RNNSearch, ConvS2S and Transformer. The comparison methods are base (without data augmentation), Edunov (reverse translation method), Fadaee (low-frequency word replacement method), our method and our method (+GEC). The experimental results of each model method in the rich- and low-resource scenarios are shown in Tables 6 and 7.

Table 6. Experimental results of English–German translation under the rich-resource scenario.

Backbone Network	Model Method	BLEU Score of Validation Set	BLEU Score of Test Set
		Newstest2014 + Newstest2015	Newstest2016
RNNSearch	Base	20.90	22.42
	Edunov	23.13	24.59
	Fadaee	21.77	23.74
	Our method	**23.69**	**24.82**
ConvS2S	Base	22.82	25.15
	Edunov	26.34	27.19
	Fadaee	25.15	25.92
	Our method	**27.58**	**28.43**
Transformer	Base	24.1	27.31
	Edunov	29.67	32.09
	Fadaee	27.94	30.50
	Our method	**30.12**	**32.61**

Table 7. Experimental results of English–German translation in low-resource scenarios.

Backbone Network	Model Method	BLEU Score of Validation Set	BLEU Score of Test Set
		Newstest2014 + Newstest2015	Newstest2016
RNN-Search	Base	13.90	14.62
	Edunov	16.19	16.87
	Fadaee	14.08	15.82
	Our method	16.41	16.33
	Our method (+GEC)	**16.71**	**17.20**
ConvS2S	Base	14.51	15.71
	Edunov	16.80	17.49
	Fadaee	15.89	16.60
	Our method	16.36	17.72
	Our method (+GEC)	**17.12**	**18.11**
Trans-former	Base	14.68	16.10
	Edunov	16.91	18.21
	Fadaee	16.09	17.32
	Our method	16.57	17.90
	Our method (+GEC)	**17.29**	**18.91**

When comparing the results of data augmentation methods from Base, Edunov, Fadaee, and our method in a resource-rich scenario, it is evident that the bilingual corpus generated using our proposed data augmentation method, which combines low-frequency word replacement and back-translation, achieved the highest translation evaluation scores across three different backbone neural machine translation models. Our method outperformed both the baseline model that did not use any data augmentation methods and the models that utilized only a single data augmentation method (Edunov and Fadaee). Compared to Edunov, the champion of the WMT2018 English-to-German translation task, our method improved BLEU scores by 0.23, 1.24, and 0.52 on the newstest2016 test set.

In a low-resource scenario, when comparing data augmentation methods from Base, Edunov, Fadaee, and our method, Edunov performed the best, while our method yielded inferior results to Edunov. However, after the pseudo-parallel corpus underwent grammar correction using the GEC module, our method (+GEC) achieved the best performance, demonstrating the necessity of grammar correction for pseudo-corpus in low-resource scenarios. In low-resource scenarios, neural machine translation models are more sensitive to the quality of the corpus, and grammar errors introduced by word replacement methods can further compromise translation performance. In contrast, in a resource-rich scenario, some grammar errors will not affect translation performance and can even enhance the robustness of the model's encoder, which is consistent with the findings of Edunov et al. This study proposes a solution to this phenomenon through the grammar correction module.

3.5. Analysis of Results

For other experimental results in Tables 5 and 6, further analysis will be combined with the charts here.

(1) Comparative analysis of backbone network

In Figure 6, the Transformer model outperforms the RNNSearch and ConvS2S models in terms of overall translation performance, regardless of whether it is in a rich- or low-resource scenario. However, the performance difference between the three models is more significant in the rich-resource scenario, with the Transformer model having the highest translation quality, followed by the ConvS2S model and then the RNNSearch model. In contrast, in the low-resource scenario, the performance difference between the three models is not as obvious.

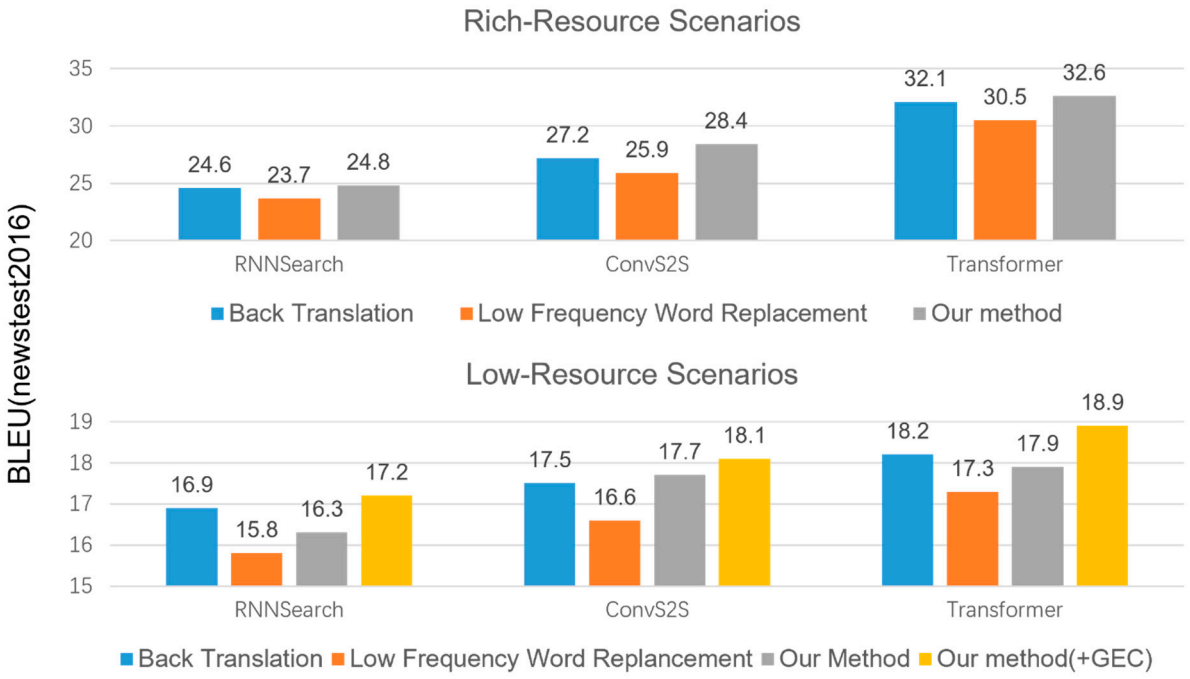

Figure 6. Translation performance of different translation models using different data augmentation methods in different scenarios.

(2) Comparative analysis of data enhancement methods

Comparing the results of the Edunov and Fadaee methods shows that both data augmentation methods improve the translation performance, which explains that neural machine translation is data-driven and the size of the parallel corpus directly affects translation performance. Among them, the translation performance improvement brought

by the back-translation method is generally higher than that of the low-frequency word replacement method. In the resource-rich scenario, Edunov based on the back-translation method outperformed Fadaee based on the low-frequency word replacement method by 0.85, 1.27, and 1.59 BLEU points on the newstest2016 test set, while Fadaee based on the low-frequency word replacement method outperformed the base method without data augmentation by 1.32, 0.77, and 3.19 BLEU points, respectively. In the low-resource scenario, Edunov outperformed Fadaee by 1.05, 0.89, and 0.89 BLEU points, respectively, on the newstest2016 test set, while Fadaee outperformed the base method by 1.20, 0.89, and 1.22 BLEU points, respectively. The reason why the back-translation method brings a higher improvement in translation performance compared to the word replacement method is that the back-translation method essentially uses additional monolingual data to generate pseudo-parallel corpora, while the word replacement method only replaces words on the basis of the original bilingual parallel corpus to generate new data.

(3) Corpus size and translation performance

During the experiment, the scale changes in the corpus after applying different data enhancement strategies were counted, as shown in Table 8.

Table 8. Corpus size under different data augmentation methods.

Model Method	Data Scale	
	Rich-Resource Scenarios	**Low-Resource Scenarios**
Original Bilingual Corpus	3.9 M	400 k
+Back Translation Method	5.9 M	2.4 M
+Low Frequency Word Replacement Method	5.2 M	720 k
Our Method	8.9 M	4.6 M

Here, the control variable method is used for analysis, the backbone network is selected as the Transformer model, and the results of the newstest2016 test set are selected as indicators. Figure 7 shows the relationship between the corpus size and the corresponding BLEU.

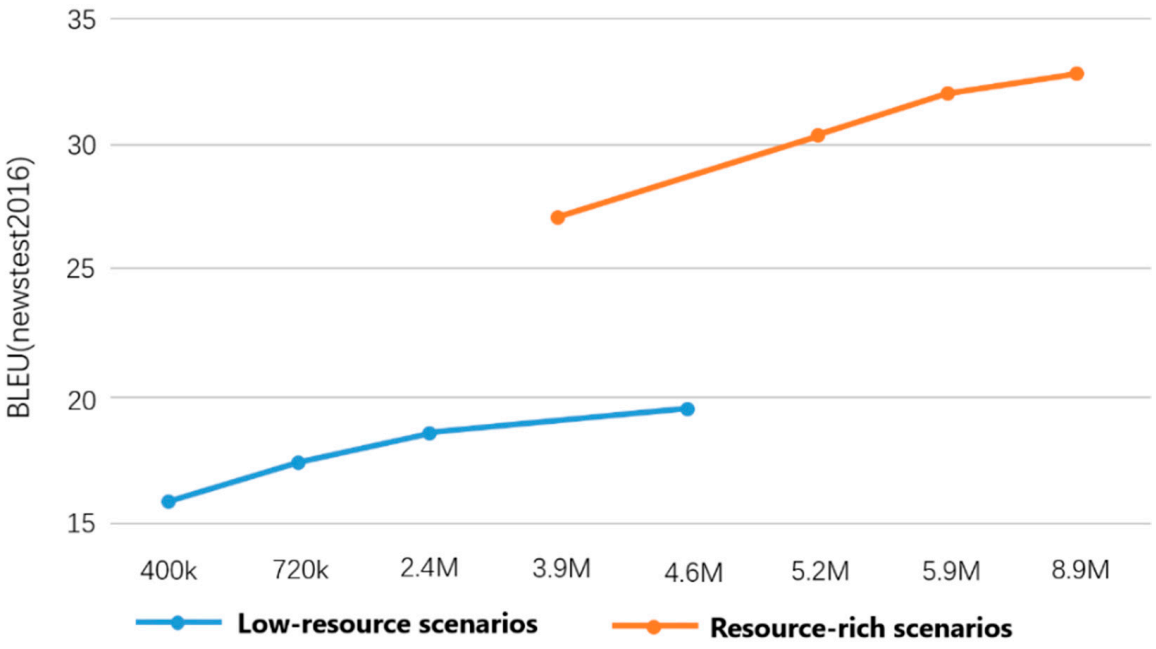

Figure 7. The relationship between the size of the corpus and the change of BLEU value.

From Figure 7, we can further verify the dependence of neural machine translation on data. Whether it is a low-resource scenario or a rich-resource scenario, when the scale of training data increases, the BLEU value on the newstest2016 test set also increases.

However, the trend in increasing BLEU scores has gradually slowed down. This means that the performance improvement for translation tasks using pseudo-bilingual corpora constructed through manual data augmentation methods is limited and cannot be infinitely increased. The core factor that affects translation performance remains to be the size of real bilingual corpora. For instance, in a rich-resource scenario with a real parallel corpus size of 3.9 M sentence pairs, the overall BLEU score for translation tasks is over 10 points higher compared to a low-resource scenario with a real parallel corpus size of 400 k.

4. Discussion

The method proposed in this paper combines the back translation method with the low-frequency word replacement method in an organic way. This two-dimensional approach enhances data in a manner that could lead to a "1 + 1 > 2" effect for strictly data-driven neural machine translation. Moreover, because the neural machine translation model has different requirements for artificially constructed pseudo-parallel corpora in low-resource and rich-resource scenarios, higher-quality pseudo-parallel corpora in low-resource scenarios can improve translation performance even more, while a more diverse pseudo-parallel corpus in rich-resource scenarios can lead to further performance improvements. This study introduces more diversity into the training data by replacing common words with less frequent words, allowing the model to learn to translate a wider range of words and phrases and improving the accuracy of translations. The method also helps prevent models from overfitting limited training datasets by adding more diverse training examples. The grammar error correction module reduces errors in the generated corpus, improving the quality of the training data and helping the model learn correct grammar and syntax, resulting in more accurate translations. Finally, merging the augmented corpus with the original parallel corpus ensures that the model is trained on a balanced combination of augmented and original data, improving its ability to handle both common and rare words and phrases. Therefore, this study proposes a data augmentation method that is suitable for both low-resource and rich-resource scenarios and can dynamically satisfy corpus quality and diversity. In low-resource scenarios, the low-frequency word replacement method enhances the corpus based on the pseudo-parallel corpus generated by the reverse translation method, and the GEC module reduces grammatical errors and ensures the quality of the enhanced corpus. In rich-resource scenarios, low-frequency word substitution increases the diversity of pseudo-parallel corpora.

5. Conclusions

The main content of this paper is to propose a general neural machine translation data enhancement method for scenarios, which combines the low-frequency word replacement method and the reverse translation method to further enhance the training corpus. It has better results in low-resource scenarios.

This paper first introduces one of the challenges faced by current neural machine translation—data augmentation—and then proposes a combined method, which is a data augmentation method that combines the low-frequency word replacement method and the reverse translation method. Then, related technologies such as the reverse translation method, low-frequency word replacement method and grammatical error correction module are introduced. Finally, the overall process and specific connotation of the combined data enhancement method proposed in this paper are introduced.

The experimental part uses the WMT2015 English–German dataset and divides the rich- and low-resource scenarios and conducts comparative experiments from two perspectives. One is the comparative experiment of different backbone networks. The comparison methods include RNNSearch, ConvS2S, and Transformer, a comparison between related data augmentation works, including Edunov, Fadaee, and our method. Ultimately, the

results in both rich- and low-resource scenarios collectively show that the combined method proposed in this paper can effectively augment the data and outperform related works.

Author Contributions: Conceptualization, W.Z. and L.Y.; software J.H. and M.L.; validation, Z.Y.; formal analysis, X.L. and Z.Y.; investigation, J.H.; resources, M.L. and Z.Y.; writing—original draft preparation, X.L., J.H., Z.Y. and L.Y.; writing—review and editing, Z.Y., M.L., W.Z. and L.Y.; visualization, Z.Y. and J.H.; supervision, W.Z.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by Sichuan Science and Technology Program (2021YFQ0003, 2023YFSY0026, 2023YFH0004).

Data Availability Statement: Data are available at EMNLP 2015 10th Workshop on Statistical Machine Translation, accession numbers: WMT2015, link is http://www.statmt.org/wmt15/translation-task.html (accessed on 1 April 2023). Data are available at ACL 2019 4th Conference on Machine Translation, accession numbers: WMT2019, link is http://www.statmt.org/wmt19/translation-task.html (accessed on 1 April 2023).

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

References

1. Stahlberg, F. Neural machine translation: A review. *J. Artif. Intell. Res.* **2020**, *69*, 343–418. [CrossRef]
2. Dabre, R.; Chu, C.; Kunchukuttan, A. A survey of multilingual neural machine translation. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–38. [CrossRef]
3. Ranathunga, S.; Lee, E.-S.A.; Prifti Skenduli, M.; Shekhar, R.; Alam, M.; Kaur, R. Neural machine translation for low-resource languages: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–37. [CrossRef]
4. Klimova, B.; Pikhart, M.; Benites, A.D.; Lehr, C.; Sanchez-Stockhammer, C. Neural machine translation in foreign language teaching and learning: A systematic review. *Educ. Inf. Technol.* **2023**, *28*, 663–682. [CrossRef]
5. Wan, Y.; Yang, B.; Wong, D.F.; Chao, L.S.; Yao, L.; Zhang, H.; Chen, B. Challenges of neural machine translation for short texts. *Comput. Linguist.* **2022**, *48*, 321–342. [CrossRef]
6. Liu, Y.; Zhang, M. Statistical machine translation. In *Routledge Encyclopedia of Translation Technology*; Routledge: London, UK, 2023; pp. 208–218.
7. Zhang, Z.; Poguda, A. Research on the Development of Data Augmentation Techniques in the Field of Machine Translation. *Int. J. Open Inf. Technol.* **2023**, *11*, 33–40.
8. Xiao, Y.; Liu, L.; Huang, G.; Cui, Q.; Huang, S.; Shi, S.; Chen, J. BiTIIMT: A bilingual text-infilling method for interactive machine translation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 1958–1969.
9. Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
10. Bahdanau, D.; Cho, K.H.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
11. Luong, T.; Pham, H.; Manning, C.D. *Effective Approaches to Attention-Based Neural Machine Translation*; Association for Computational Linguistics: Lisbon, Portugal, 2015; pp. 1412–1421. [CrossRef]
12. Liu, L.; Utiyama, M.; Finch, A.; Sumita, E. *Agreement on Target-Bidirectional Neural Machine Translation*; Association for Computational Linguistics: San Diego, CA, USA, 2016; pp. 411–416. [CrossRef]
13. Zhou, L.; Zhang, J.; Zong, C. Synchronous Bidirectional Neural Machine Translation. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 91–105. [CrossRef]
14. Xiong, H.; He, Z.; Hu, X.; Wu, H. Multi-Channel Encoder for Neural Machine Translation. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*. [CrossRef]
15. Shen, S.; Cheng, Y.; He, Z.; He, W.; Wu, H.; Sun, M.; Liu, Y. *Minimum Risk Training for Neural Machine Translation*; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1683–1692. [CrossRef]
16. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. *Hierarchical Attention Networks for Document Classification*; Association for Computational Linguistics: San Diego, CA, USA, 2016; pp. 1480–1489. [CrossRef]
17. Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; Li, H. Modeling Coverage for Neural Machine Translation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 76–85.
18. Tu, Z.; Liu, Y.; Shang, L.; Liu, X.; Li, H. Neural machine translation with reconstruction. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Fransico, CA, USA, 4–9 February 2017; AAAI: San Fransico, CA, USA, 2017.
19. Nguyen, X.-P.; Joty, S.; Wu, K.; Aw, A.T. Data diversification: A simple strategy for neural machine translation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 10018–10029.
20. Xie, S.; Xia, Y.; Wu, L.; Huang, Y.; Fan, Y.; Qin, T. End-to-end entity-aware neural machine translation. *Mach. Learn.* **2022**, *111*, 1181–1203. [CrossRef]

21. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
22. Abdulmumin, I.; Galadanci, B.S.; Isa, A. Enhanced back-translation for low resource neural machine translation using self-training. In Proceedings of the Information and Communication Technology and Applications: Third International Conference, ICTA 2020, Revised Selected Papers 3, 2021, Minna, Nigeria, 24–27 November 2020; pp. 355–371.
23. Dijkstra, T.; Wahl, A.; Buytenhuijs, F.; Van Halem, N.; Al-Jibouri, Z.; De Korte, M.; Rekké, S. Multilink: A computational model for bilingual word recognition and word translation. *Biling. Lang. Cogn.* **2019**, *22*, 657–679. [CrossRef]
24. Sugiyama, A.; Yoshinaga, N. Data augmentation using back-translation for context-aware neural machine translation. In Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019), Hong Kong, China, 3 November 2019; pp. 35–44.
25. Edunov, S.; Ott, M.; Auli, M.; Grangier, D. *Understanding Back-Translation at Scale*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 489–500. [CrossRef]
26. Fadaee, M.; Bisazza, A.; Monz, C. *Data Augmentation for Low-Resource Neural Machine Translation*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 567–573. [CrossRef]
27. Sennrich, R.; Haddow, B.; Birch, A. *Neural Machine Translation of Rare Words with Subword Units*; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1715–1725. [CrossRef]
28. Dyer, C.; Chahuneau, V.; Smith, N.A. A simple, fast, and effective reparameterization of IBM model 2. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 644–648.
29. Zhao, W.; Wang, L.; Shen, K.; Jia, R.; Liu, J. *Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 156–165. [CrossRef]
30. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
31. Durrani, N.; Sajjad, H.; Hoang, H.; Koehn, P. Integrating an unsupervised transliteration model into statistical machine translation. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 26–30 April 2014; Volume 2, pp. 148–153.
32. Zhang, Z. *Advanced Data Augmentation Strategy for Neural Machine Translation*; Universiry of Science and Technology of China: Chendu, China, 2019.
33. Myle, O.; Michael, A.; David, G.; Marc'Aurelio, R. Analyzing Uncertainty in Neural Machine Translation. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Jennifer, D., Andreas, K., Eds.; PMLR, Proceedings of Machine Learning Research: Stockholm, Sweden, 2018; Volume 80, pp. 3956–3965.
34. Pascal, V.; Hugo, L.; Yoshua, B.; Pierre-Antoine, M. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Association for Computing Machinery, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
35. Ge, T.; Wei, F.; Zhou, M. *Fluency Boost Learning and Inference for Neural Grammatical Error Correction*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 1055–1065. [CrossRef]
36. Alokla, A.; Gad, W.; Nazih, W.; Aref, M.; Salem, A.-B. Retrieval-Based Transformer Pseudocode Generation. *Mathematics* **2022**, *10*, 604. [CrossRef]
37. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: Sydney, Australia, 2017; pp. 1243–1252.
38. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *Comput. Sci.* **2014**. [CrossRef]