applied sciences

MDPI

Article

Boosting the Transformer with the BERT Supervision in Low-Resource Machine Translation

Rong Yan, Jiang Li, Xiangdong Su *, Xiaoming Wang and Guanglai Gao

College of Computer Science, Inner Mongolia University, Hohhot 010020, China; csyanr@imu.edu.cn (R.Y.); futuretopdelli@163.com (J.L.); 13488208986@163.com (X.W.); csggl@imu.edu.cn (G.G.)
* Correspondence: cssxd@imu.edu.cn

Abstract: Previous works trained the Transformer and its variants end-to-end and achieved remarkable translation performance when there are huge parallel sentences available. However, these models suffer from the data scarcity problem in low-resource machine translation tasks. To deal with the mismatch problem between the big model capacity of the Transformer and the small parallel training data set, this paper adds the BERT supervision on the latent representation between the encoder and the decoder of the Transformer and designs a multi-step training algorithm to boost the Transformer on such a basis. The algorithm includes three stages: (1) encoder training, (2) decoder training, and (3) joint optimization. We introduce the BERT of the target language in the encoder and the decoder training and alleviate the data starvation problem of the Transformer. After the training stage, the BERT will not further attend the inference section explicitly. Another merit of our training algorithm is that it can further enhance the Transformer in the task where there are limited parallel sentence pairs but large amounts of monolingual corpus of the target language. The evaluation results on six low-resource translation tasks suggest that the Transformer trained by our algorithm significantly outperforms the baselines which were trained end-to-end in previous works.

Keywords: transformer; latent representation; machine translation; low-resource; BERT

Citation: Yan, R.; Li, J.; Su, X.; Wang, X.; Gao, G. Boosting the Transformer with the BERT Supervision in Low-Resource Machine Translation. *Appl. Sci.* **2022**, *12*, 7195. https://doi.org/10.3390/app12147195

Academic Editors: Valentino Santucci and Douglas O'Shaughnessy

Received: 7 May 2022
Accepted: 14 July 2022
Published: 17 July 2022

1. Introduction

With the development of deep learning, neural machine translation (NMT) [1,2] makes significant progress and outperforms the statistical machine translation on the language pairs with an abundance of the parallel corpus. Among these NMT models, the Transformer [3] is well known for producing state-of-the-art (SOTA) performance in many translation tasks [4–6]. The Transformer consists of a multi-layer encoder and a multi-layer decoder. The encoder reads the source language sequence and maps it into a fixed-length representation, and the decoder decodes the fixed-length representation and outputs the target language sequence. Previous studies trained the encoder and the decoder synchronously in an end-to-end fashion.

However, the Transformer suffers in low-resource translation tasks [7–11] where there is not a large-scale parallel corpus available. The core of this problem is the mismatch between the big model capacity and the small training parallel data available. When there is only a small scale of translation instances available, it is challenging to optimize the parameters of the Transformer very well, leading to the unsuitable representation of the input sentence from the encoder and the undesired translation result from the decoder. The reason is that training the big translation model in the end-to-end fashion with a small training dataset will lead to a severe overfitting problem. Therefore, how to alleviate the overfitting problem and further improve the Transformer in low-resource machine translation is garnering much attention.

To better optimize the parameters of transformers in low-resource machine translation and prevent the overfitting problem, we design a three-stage training approach

as an alternative to the traditional end-to-end training method. Specifically, we add the BERT supervision on the latent representation between the encoder and the decoder of the Transformer. On such basis, we train the encoder and the decoder independently to alleviate the mismatch problem between the model capacity and the training data since the facts are Capacity(Encoder) $\ll$ Capacity(Transformer) and Capacity(Decoder) $\ll$ Capacity(Transformer). The overall training algorithm includes three stages: (1) encoder training, (2) decoder training, and (3) joint optimization. Joint optimization fine tunes the parameters to make the encoder and the decoder better matched in the latent space.

First, the BERT is the fine-tuning-based representation model that can produce accurate contextual embeddings of sentences [12]. Based on this fact, we argue that the BERT can be treated as a good representation of the semantic space and try to make the encoded representation of the source sentences align with their target sentences in the BERT representation when training the Transformer, as shown in Figure 1. An exemplary encoding representation will benefit the model convergence and final performance of the Transformer. A recent work [13] proposed a multilingual Transformer which implicitly learned shared representation of many different languages in the semantic space. Sennrich et al. [14] explored the strategy to include monolingual training data in the training process without changing the model structure and found that using synthetic data to fill the source side is more effective.

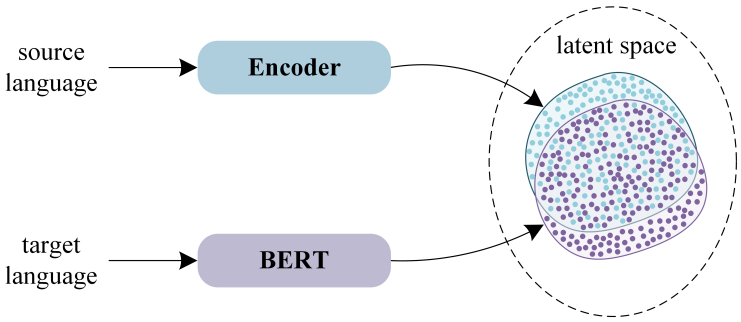

Figure 1. The principle of our approach: aligning the source language and the target language in latent space.

Second, adding BERT supervision enables us to independently train the encoder and the decoder of the Transformer (shown in Figure 2), which is different from the previous works that trained the Transformer end-to-end. According to the neural network theory, the model capacity is directly related to the number of hidden units, the layer depth, and the operations [15]. The model capacity of the entire Transformer is bigger than that of its encoder or its decoder. If there is only a small-scale training dataset, smaller models are less prone to overfitting. Therefore, it is reasonable to believe that independently training the encoder and decoder in the Transformer helps mitigate the mismatch problem between the big model capacity of the Transformer and the small parallel training dataset and thus contribute to a better-trained Transformer than that trained end-to-end.

Third, training the decoder of the Transformer uses the monolingual corpus of the target language. Provided the scenario of low parallel corpus but high monolingual target language corpus, a byproduct of BERT supervision is that we can further improve the translation performance by training the decoder with the large-scale monolingual corpus of the target language.

Although Zhu et al. [16] used the BERT in machine translation, there are two critical differences between our approach and the BERT-fused model. First, our approach does not change the structure of the Transformer. The parameter of the Transformer trained with our algorithm is far less than that of the BERT-fused model. Second, the BERT-fused model exploits the representation from the BERT by feeding it into all layers, while we use the BERT as the learning target of the encoder. The Transformer trained with our algorithm infers faster than the BERT-fused model. Our approach also differs from the previous

machine translation approaches, which used a pre-trained language model to improve the encoder and decoder of NMT [17–19]. The contributions of this paper are as follows:

- We boost the Transformer by adding the BERT constraint on the latent representation in low-resource machine translation. As such, we design a training algorithm to optimize the Transformer in a multi-step way, including encoder training, decoder training, and joint optimization. It alleviates the mismatch problem between the capacity of the Transformer and the training data size in low-resource machine translation.
- We provide a new way to incorporate the pre-trained language models in machine translation. Compared to the BERT-fused methods, a significant advantage of our approach is that it improves the performance of the Transformer in low-resource translation tasks without changing its structure and increasing the number of parameters.
- Adding BERT supervision enables us to further improve the Transformer with a large-scale monolingual target language dataset in the sense of a low parallel corpus but a high monolingual target language corpus.

The remainder of this paper is structured as follows. Section 2 describes related work. Section 3 introduces our approach. Section 4 describes the experiment setting. Section 5 reports the results and the analysis. Finally, Section 6 presents our conclusions.

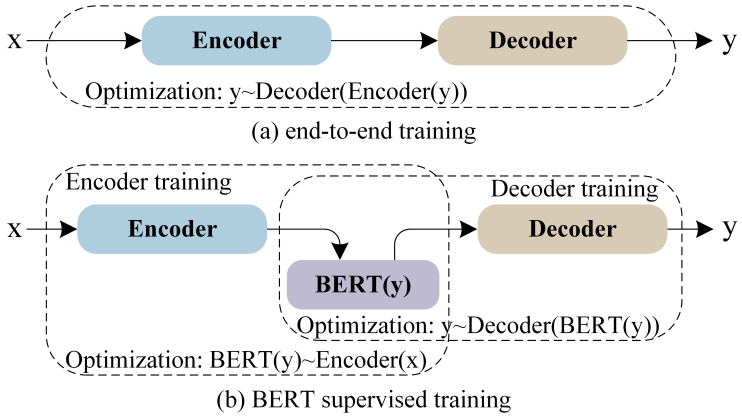

Figure 2. Illustration of end-to-end training and BERT supervised training of the Transformer. The facts are Capacity(Encoder) $\ll$ Capacity(Transformer) and Capacity(Decoder) $\ll$ Capacity(Transformer).

2. Related Work

All related works are listed in Table 1. They are divided into the following categories.

Table 1. Comparison of the related models.

Category	Models	Key Idea
Transformer and its variants	NMT-JL [20], Seq2Seq [1]	Advancing the translation quality via neural networks and attention mechanisms.
	Attention is all you need [3]	It relies entirely on an attention mechanism to draw global dependencies between sources.
	GRET [21]	A novel global representation enhanced the Transformer to model the global representation explicitly in the Transformer.
	Transparent [22]	The encoder layers were combined just after the encoding is completed but not during the encoding process.
	DLCL [23]	An approach based on a dynamic linear combination of layers to memorizing the features extracted from all preceding layers.

Table 1. *Cont.*

Category	Models	Key Idea
MT Methods with Pre-training Language Models	Elmo [24], Xlnet [25], Roberta [26], GPT [27]	The pre-training language models are effective for the machine learning.
	BERT [12]	Designing the BERT to pre-train deep bidirectional encoder representations from unlabeled text to produce contextualized embedding.
	MASS [17]	Adopting the encoder–decoder framework to reconstruct a sentence fragment with the remaining part of the sentence.
	CT_{NMT} [28]	Integrating the pre-trained LMs to neural machine translation.
	NMT-BERT [19]	The pre-trained models should be exploited for supervised neural machine translation.
	BERT-fused [16]	Using BERT to extract representations for input sequences. Then the representations are fused with each layer of the NMT model through attention mechanisms.
Low-resource Machine Translation Models	NMT-TL [29]	Proposing transfer learning for NMT.
	NMT-EPL [30]	Utilizing English as a bridging language.
	DLMT [31]	Dual-learning mechanism for machine translation.
	NMT-RT [32]	A new round-tripping approach.
	NMT-Attention [33]	An unsupervised method based on an attentional NMT system.
	NMT-lexically aligned [34]	Optimizing the cross-lingual alignment of word embeddings on unsupervised Macedonian–English and Albanian–English.
	NMT-regularization factors [35]	Exploring the roles and interactions of the hyperparameters governing regularization.

2.1. Transformer and Its Variants

Neural machine translation (NMT) models advance the translation quality via neural networks and attention mechanisms [1,20]. Among these models, the Transformer reaches a new state of the art. It relies entirely on an attention mechanism to draw global dependencies between source and target and achieves strong results on several large-scale tasks, dispensing with recurrence and convolutions [3]. Weng et al. [21] designed a novel global representation that enhanced the Transformer (GRET) to model the global representation explicitly in the Transformer. The encoder generated an external state for the global representation, which was then fused into the decoder during the decoding process to improve generation quality. Bapna et al. [22] pointed out that the vanilla Transformer was hard to train if the depth of the encoder was beyond 12. They successfully trained a 16-layer encoder by attending the combination of all encoder layers to the decoder. In their approach, the encoder layers were combined just after the encoding was completed but not during the encoding process. To make the Transformer deeper, Wang et al. [23] propose an approach based on a dynamic linear combination of layers to memorize the features extracted from all preceding layers. They demonstrate that layer normalization is helpful to learning deep encoders. The encoder was optimized smoothly by relocating the layer normalization unit. However, the above works mainly focus on the translation tasks with numerous sentence pairs, and the models were trained in an end-to-end fashion.

2.2. MT Methods with Pre-Training Language Models

Previous works have shown that the pre-training language models [24–27] are effective for the machine learning task. Jacob Devlin et al. [12] designed the BERT to pre-train deep bidirectional encoder representations from unlabeled text to produce contextualized embedding. The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models in semantic tasks. Kaitao Song et al. [17] adopted the encoder–decoder framework to reconstruct a sentence fragment with the remaining part of the sentence by masked sequence-to-sequence pre-training. Their approach achieved state-of-the-art accuracy on the unsupervised English–French translation. To avoid the catastrophic forgetting, Jiacheng Yang et al. [28] proposed the CT_{NMT} to integrate the pre-trained LMs to neural machine translation. Clinchant et al. [19] studied how the pre-trained models should be exploited for supervised neural machine translation. They compared various ways to integrate the pre-trained BERT model with the NMT model and studied the impact of the monolingual data used for the BERT training on the final translation quality. Zhu et al. [16] proposed an algorithm named the BERT-fused model, in which the BERT is used to extract representations for an input sequence. Then the representations are fused with each layer of the NMT model through attention mechanisms. All the above methods increased the model parameters while lowering the translation speed.

2.3. Low-Resource Machine Translation Models

The lack of parallel data is challenging for NMT model training. Qi et al. [36] indicated that pre-trained embeddings are particularly effective in low-resource environments. Zoph et al. [29] first employed transfer learning for NMT. Specifically, they utilized the trained parent model parameters to initialize a child model, and then trained on the desired low-resource pair. Sennrich et al. [14] used monolingual training data during training of NMT systems for the low-resource NMT task. In addition, parallel data also can be included in these pre-training approaches [37–39]. Ahmadnia et al. [30] utilized English as a bridging language to improve the quality of the Persian–Spanish low-resource. DLMT [31] proposed a dual-learning mechanism for machine translation, e.g., English to French translation (primal) versus French to English translation (dual). Through the dual-learning process, one agent represents the model of the primal task while the other represents the model of the dual task, then we ask them to communicate and learn from each other through reinforcement learning. Ahmadnia et al. [32] applied a new round-tripping approach that incorporates dual learning [31] for automatic learning from unlabeled data but transcends prior work through effective leveraging of monolingual text. Xu et al. [33] proposed an unsupervised method based on an attentional NMT system for Spanish–Turkish low-resource. Chronopoulou et al. [34] optimized the cross-lingual alignment of word embeddings on unsupervised Macedonian–English and Albanian–English. The recent work [35] analysed the roles and interactions of the hyperparameters governing regularization and presented a range of values applicable to low-resource NMT.

3. Approach

3.1. Latent Representation Using the BERT

In the Transformer, the encoder turns each input token into one embedding vector and generates the hidden representations through the attention mechanisms, and the decoder maps the hidden representations in the latent space into the target sentence. In previous works, the Transformer was trained end-to-end without any restriction on the latent space. Only the dimension of the latent representation was treated as a hyperparameter and optimized through the validation experiments. In a low-resource translation task, end-to-end training of the Transformer will very likely lead to ineligible representations. Since the BERT has achieved great success in language understanding tasks, we argue that it can be treated as a good representation of the semantic space. A good representation will benefit the model convergence and final performance. Thus, this paper selects the BERT as

the latent representation of the encoder and makes the source sentences and their target sentences align in this specific space. The reasons are as follows:

First, previous works have shown that language model pre-training is effective for improving many natural language processing tasks. Sascha Rothe et al. [40] used the BERT as the encoder and the GPT2 as the decoder to form an encoder–decoder framework. It proved that an ideal encoder is good for the performance. Second, the BERT is the first fine-tuning-based representation model that produces contextual embeddings and achieves state-of-the-art performance on an extensive suite of sentence-level tasks. It is a subtle and accurate representation of the word in a sentence. Third, the BERT is a variant of the Transformer architecture, which is similar to the encoder in the Transformer.

By adding the BERT constraint on the latent representation, we make it possible to train the encoder and the decoder of the Transformer independently. Let x be the source sentence, and y be the corresponding target sentence, then the learning target of the encoder on x is the output of the BERT on y in the encoder training. The decoder of the Transformer is to map the latent representation into the target language. In decoder training, the input is the representation of y from the BERT, and the output is y.

We introduce the BERT of the target language to generate the training data set from the parallel corpus for the encoder and the decoder training. That is, the training data sets for the encoder and the decoder training are in the form of (x, the BERT(y)) and (the BERT(y), y), respectively. We describe the training algorithm in the following section.

3.2. Training Algorithm

According to the motivation above, we design a training algorithm to optimize the Transformer, as shown in Algorithm 1. The training algorithm includes three stages: (1) encoder training, (2) decoder training, and (3) joint optimization. Concerning the fact that independently training the encoder and the decoder cannot make the encoder and decoder fully match in the latent space, we add the joint optimization in the training algorithm to further fine tune the encoder and decoder in a few epochs. The subsequent experiments also prove that joint optimization is good for performance.

Algorithm 1 The Training Algorithm

Data set construction:

- Parallel corpora set $A = \{(x_1, y_1), \ldots (x_n, y_n)\}$
- Compute $theBERT(y_i)$ for each y_i in A
- Build the dataset B and C

  $B = \{(x_1, theBERT(y_1)), \ldots (x_n, theBERT(y_n))\}$

  $C = \{(theBERT(y_1), y_1), \ldots (theBERT(y_n), y_n)\}$

(1) Training Encoder using dataset B

(2) Training Decoder using dataset C

(3) Joint optimization for Encoder and Decoder using data set A

Let X and Y be the sentences of the source language and the target languages, respectively, t is the Transformer, g is the encoder in the Transformer, and f is the decoder in the Transformer. We represent the translation process as $Y = t(X) = f(g(X))$. The capacity of the Transformer t is larger than that of the encoder g or the decoder f. When only a small training data set is available, the smaller models are less likely to overfit the training data. Therefore, it is reasonable to believe that independently training the encoder and decoder of the Transformer helps mitigate the mismatch problem between the training data set and the model capacity and will end up with a translation model which performs better than that trained directly in an end-to-end fashion. We generate the training datasets for encoder training and decoder training in the preprocessing step.

3.2.1. Encoder Training

As shown in Algorithm 1, the encoder and the BERT map the source language x and the target language y into latent space, respectively. The learning goal of the encoder is the latent representation from the BERT $Bert_y$. The input of the encoder is the source language I_x, and its output is $Enc_x = Encoder(I_x)$. Meanwhile, the BERT maps the input target language y into $Bert_y$: $Bert_y = BERT(y)$. We extend the length of each source sentence and target sentence to *512* by adding *pad* tokens.

The encoder training objective is to penalize the mean-squared error MSE loss between $Bert_y$ and Enc_x:

$$L_{enc} = \| \ Enc_x - Bert_y \ \|_2^2 \tag{1}$$

In Equation (1), both the output of the encoder and the BERT are three-dimensional matrices.

The structure of the encoder is from the Transformer [3], which consists of six stacked layers. Each layer comprises two sub-layers, namely a multi-head self-attention layer and a fully connected feed-forward layer. Each sub-layer has residual connection and normalization.

3.2.2. Decoder Training

As mentioned, machine translation can be divided into the encoding stage and the decoding stage. The decoder maps the latent representations into the target sentences in the decoding stage. Since we make the encoded representation of the source sentences align with their target sentences in the BERT representation when training the encoder of the Transformer, we use $Bert_y$ as the input of the decoder and y as the large output in decoder training, as

$$y \sim Dec_{Bert} = Decoder(Bert_y) \tag{2}$$

where $Bert_y$ represents the output of y from the pre-training model Bert. We expect Dec_{Bert} to be close to y in Equation (2). This forms an autoencoder similar network. We assemble Dec_{Bert} and y together to train the decoder, as shown in Algorithm 1.

The decoder of our network is from the Transformer [3]. Each decoder consists of six stacked layers. Each layer contains three sub-layers. Different from that of the encoder, the sub-layers of the decoder add a masked multi-head self-attention mechanism.

3.2.3. Joint Optimization

Joint optimization further makes the encoder and decoder fully matching in the latent space in a few epochs. After the first two stages, the encoder and decoder are joined together and optimized using the parallel corpus. The fine-tuned network is used in the translation tasks from the source language I_x to the target language y. The encoder maps I_x into Enc_x, as described

$$Enc_x = Encoder(I_x) \tag{3}$$

The decoder maps Enc_x into Dec_y, as described

$$Dec_y = Decoder(Enc_x) \tag{4}$$

The joined encoder–decoder network is fine-tuned to map I_x into y,

$$y \sim Dec_y = Decoder(Encoder(I_x)) \tag{5}$$

4. Experimental Setting

4.1. Data and Metric

In the experiments, we evaluate our approach on six low-resource translation tasks, including German→English (De→En), Romanian→English (Ro→En), Vietnamese→English (Vi→En), German→Chinese (De→Zh), Korean→Chinese (Ko→Zh), and Russian→Chinese (Ru→Zh). The first three tasks (the target language is English) are IWSLT' 14 De→En, WMT' 2016 Ro→En, and IWSLT' 14 Vi→En tasks, which had $160K$, $130K$, and $600K$ parallel

sentences in training data sets, respectively. We use *tst*2010, *tst*2011, and *tst*2012 as the test set for De→En translation, *newstest*2016 as the test set for Ro→En translation, and *tst*2014 as the test set for Vi→En translation, respectively. The last three tasks (the target language is Chinese) use the TED parallel corpus [36] containing De→Zh, Ko→Zh, and Ru→Zh. The three training sets contain 140*K*, 160*K*, and 130*K* sentence pairs, respectively. Each of the three testing sets contains 3000 sentences different from that in the training sets.

The evaluation metric is case-insensitive BLEU calculated by the multi-bleu.perl script [41]. BLEU is

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n p_n\right) \tag{6}$$

$$\text{BP} = \begin{cases} 1, & c > r \\ e^{1-r/c}, & c \leq r \end{cases} \tag{7}$$

where r is the source sentence, c is the translation result, N is the item number of N-gram, w_n is the coefficient of N-gram, and p_n is the precision of N-gram matching.

4.2. Implementation Details

We adopt the pre-trained BERT provided by PyTorch-Transformers [42]. For the translation tasks from other languages to English, we choose the BERT of English with 12 layers and 768 hidden dimensions. For the translation tasks from other languages to Chinese, we choose the BERT of the Chinese model with the same settings as the BERT of English. We extend the source language sentences and the target language sentences to 512 tokens with a *pad* operation to match the BERT model.

We used Adam [43] to optimize the network with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $weight - decay = 0.0001$. The initial learning rate is 0.0005 with the inverse sqrt learning rate scheduler. We set the dimension of all the hidden states in the Transformer trained with our algorithm to 768 to match the output of the pre-trained BERT model. Due to the parameter amount of the BERT, we set the batch size to 256 during the model training. In the inference stage, we set the beam width to 5 and length penalty to 0.6 following [3].

For other languages to English translation tasks, we performed the following operations on all data: (a) lower casing and accent removal, (b) punctuation splitting, and (c) white space tokenization. We preprocess all sentences by BPE [44] and set the size of the sub-words to 32*K* for each language pair. For translation tasks from other languages to Chinese, we add spaces around every character for Chinese and Korean Hanja.

4.3. Baselines

This paper compares the Transformer (base) trained with our algorithm with the following approaches that trained end-to-end, including **Transformer (base)** [3], **Transformer (big)** [3], **Transformer-based on transfer learning** [29], **DeepRepre** [45], **RelPos** [46], and **BERT-fused model** [16]. Since the proposed approach is used for the Transformer training, we compare the resulting Transformer from our algorithm with the Transformer and its variants which were trained in the traditional end-to-end way. We also compare our approach with the BERT-fused model, which incorporates the BERT of the source language as additional information in the encoding and the decoding stages.

It is worth noting that we use "Ours" to represent the Transformer (base) trained with our multi-step algorithm. All the baselines are trained in the traditional end-to-end way until they are converged.

5. Results and Discussion

5.1. Comparison with Baselines

Table 2 shows the translation performances of the baselines and the Transformer (base) trained with our approach on the six low-resource translation tasks. It is clear that the Transformer (base) trained with our algorithm achieves the highest BLEU values on these

tasks. Compared with Transformer(base) trained end-to-end, the improvements achieve 4.2%, 2.5%, 6.4%, 4.6%, 3.3%, and 4.5% on the six tasks, respectively.

Table 2. The translation performance of the baselines and our approach.

Approaches		De→En	Ro→En	Vi→En	De→Zh	Ko→Zh	Ru→Zh
Transformer-based Methods	Transformer (base)	34.98	32.06	27.87	27.88	30.24	25.43
	Transformer (big)	34.85	31.98	27.82	27.79	30.28	25.35
	Transformer-based transfer learning	35.01	32.10	27.75	27.72	30.13	25.15
	RelPos	35.12	32.15	28.57	28.12	30.51	25.69
	DeepRepre	35.41	32.27	28.48	28.35	30.59	26.14
Incorporating Bert Methods	Bert-fused NMT	36.34	32.35	29.04	28.82	30.79	26.29
	Ours	**36.45**	**32.86**	**29.65**	**29.17**	**31.24**	**26.57**

Among the baselines, the Transformer (base), the Transformer (big), and the Transformer-based transfer learning produce similar performances. RelPos and DeepRepre obtain better performances than the Transformer (base), the Transformer (big), and the Transformer-based transfer learning since they extract better representation using a deep attention mechanism. The BERT-fused model improves the translation quality by incorporating the BERT semantic information in the encoding and the decoding stages.

Compared to the Transformer (base) trained end-to-end, our approach improves the performance of the Transformer (base) about 1∼2 BLEU point. Our approach is obviously better than the RelPos and DeepRepre. It suggests that adding BERT supervision alleviates the data hungry problem of the Transformer and improves its performance in low-resource machine translation. From another viewpoint, adding the BERT supervision can transfer the BERT knowledge into the Transformer and makes this model converge very well.

Table 3 shows the number of parameters and the average inference time of each model in the test stage. We use the average inference time of Transformer (base) as the standard. The inference time of any other model is a multiple of the standard. The BERT-fused model has the most parameters and the slowest inference speed. This is because it incorporates the BERT in the inference process as additional knowledge of the encoding and the decoding modules. The Transformer (big) and DeepRepre also have more parameters than the Transformer (base). Although the BERT-fused model has a good translation quality, its inference speed is very slow. The parameter number of our approach is the same as that of the Transformer (base) and far less than that of the BERT-fused model. Since our approach does not change the structure of the Transformer (base), its inference speed is faster than the BERT-fused model.

Table 3. The parameters and average inference speed between the baselines and our approach.

Approaches	Parameters	Avg. Time
Transformer (base)	**87 M**	**1.0**×
Transformer (big)	124 M	1.29×
Transformer-based transfer learning	**87 M**	**1.0**×
DeepRepre	111 M	1.25×
RelPos	**87 M**	**1.0**×
BERT-fused NMT	197 M	1.41×
Ours	**87 M**	**1.0**×

Considering the comprehensive performance of the Transformer trained with our algorithm in terms of translation quality, model parameters, and inference speed, we draw

the conclusion that the proposed training approach is effective for the Transformer in low-resource machine translation.

Table 4 shows three examples De→En, Ro→En, and Vi→En, which indicates that the proposed approach is effective.

Table 4. Examples from Tranformer(base) and our approach on De→En, Ro→En, and Vi→En.

De→En	
Source	Und warum? Weil sie Dreiecke verstehen und sich-selbst-ver stärkende geometrische Muster sind der Schlüssel um stabile Strukturen zu bauen.
Target	And why? Because they understand triangles and self-reinforcing geometric patterns are the key to building stable structures.
Transformer(base)	And why? Because they understand triangles and self-reinforcing geometric patterns, they are crucial to building stable structures.
Ours	And why? Because they understand triangles and self-reinforcing geometric patterns are key to building stable structures.
Ro→En	
Source	Ban și-a exprimat regretul că divizările în consiliu și între poporul sirian și puterile regionale "au făcut această situație de nerezolvat".
Target	He expressed regret that divisions in the council and among the Syrian people and regional powers "made this situation unsolvable".
Transformer(base)	Ban expressed regret that the divisions in the council and between the Syrian people and the regional powers "have made this situation unresolved".
Ours	Ban expressed regret that divisions in the council and between the Syrian people and regional powers "have made this situation intractable".
Vi→En	
Source	Từng đồng_tiền đều được cân_nhắc và tiền học thêm tiếng Anh và toán được đặt riêng ra bất_kể việc khoản nào phải trừ bớt đi, thường thì đó là quần_áo mới; quần_áo chúng_tôi lúc nào cũng là đồ cũ.
Target	All the dollars were allocated and extra tuition in English and mathematics was budgeted for regardless of what missed out, which was usually new clothes; they were always secondhand.
Transformer(base)	Every dollar is considered and English and math tutoring is set separately no matter what is deducted, usually new clothes; our clothes are always second-hand.
Ours	All the money was allocated, extra English and math tuition was budgeted, whatever was missed, usually new clothes; they were always second-hand.

5.2. Effectiveness of Joint Optimization

Unlike the end-to-end training methods, our algorithm trains the Transformer in a three-stage fashion, which includes (1) encoder training, (2) decoder training, and (3) joint optimization. We design joint optimization to further fine-tune the parameter in the Transformer, so that the encoder and the decoder can perfectly match in the latent space. The first two stages are necessary, and the third stage is optional in theory. Therefore, we only analyze the results with and without joint optimization in the ablation experiment.

To verify the effectiveness of this training operation, we conduct ablation studies on the above translation tasks as shown in Table 5. We compared the performances of different optimization epochs. For simplicity, we use EncT, DecT, JO to represent encoder

training, decoder training, and joint optimization, respectively. The third line "EncT + DecT" represents the model only going through the encoder training and the decoder training. We also list the performance of the Transformer (base) trained end-to-end.

Table 5. The effectiveness of joint optimization. EncT, DecT, JO represent encoder training, decoder training, and joint optimization, respectively.

	Models	De→En	Ro→En	Vi→En	De→Zh	Ko→Zh	Ru→Zh
	Transformer (base)	34.84	32.06	27.87	27.88	30.24	25.43
Ours	EncT + DecT	30.88	26.51	22.11	22.82	25.33	18.95
	EncT + DecT + JO (40 epoch)	33.77	31.19	28.12	27.58	28.71	23.23
	EncT + DecT + JO (80 epoch)	35.28	31.67	28.38	28.03	29.59	25.87
	EncT + DecT + JO (full training)	**36.38**	**32.86**	**29.65**	**29.17**	**31.24**	**26.57**

As shown in Table 5, when the Transformer only goes through the encoder training and the decoder training independently, its performance is lower than that of the Transformer (base) trained end-to-end. This is because the encoder and the decoder do not not fully match in the latent space. The performance of the Transformer goes up significantly as joint optimization epochs increase from 40 to 80. The model obtains the highest BLEU 36.38 in De→En task after the full fine-tuning, which shows that joint optimization can boost the Transformer after independently training the encoder and the decoder. With the increase of the epoch of joint optimization, the performance of the Transformer increases. The results on the other tasks also confirm this point.

5.3. Effectiveness of the BERT Fine-Tuning

In decoder training, we studied whether the parameters of the BERT model need to be frozen. To this end, we adjusted the training sequence. First, we trained the decoder of the Transformer. Then, we trained the encoder of the Transformer. Finally, we performed joint optimization in the third stage. Table 6 lists the performances of the Transformer on the six tasks.

Table 6. The effectiveness of the BERT Fine-tuning.

Strategy	De→En	Ro→En	Vi→En	De→Zh	Ko→Zh	Ru→Zh
BERT frozen	35.86	32.46	29.08	28.76	30.81	26.29
BERT fine-tuning	**36.38**	**32.86**	**29.65**	**29.17**	**31.24**	**26.57**

Obviously, compared to the case where the BERT is frozen, the BLEU of the Transformer improves when the BERT is fine-tuned in the decoder training. This is because we used the pre-training model to get good initialization parameters and further improve through fine-tuning. It suggests that the BERT fine-tuning is good for machine translation. At the same time, however, the training time and the memory will increase if the BERT is fine-tuned in the training process. From Tables 2 and 6, we prove that training the Transformer with our approach is much better than training it end-to-end no matter whether the BERT is frozen.

5.4. Effectiveness of the Large-Scale Monolingual Corpus of the Target Language

As mentioned, in the scenario of low parallel corpus but high monolingual target language corpus, a byproduct of BERT supervision is that we can further improve the translation performance by training the decoder with the large-scale monolingual corpus of the target language. To validate this point, we conducted experiments on the De→En task, in which we added the extra large-scale monolingual English corpus TED and WMT to the original training dataset IWSLT'14 in decoder training. The TED includes 500,000 English sentences from www.kaggle.com (accessed on 8 January 2020) , and the WMT includes

4,000,000 English sentences from www.statmt.org (accessed on 8 January 2020) . The results are listed in Table 7. "EncT + DecT" represents the encoder training and decoder training using the IWSLT'14; "+TED" and "+WMT" mean that we add the TED and WMT to train the dataset in decoder training.

Table 7. Effectiveness of the decoder training using an additional large-scale monolingual corpus of the target language. DecT(+TED) and DecT(+WMT) mean that we used the additional TED and WMT in decoder training, respectively.

Approaches	De→En
EncT + DecT	30.88
EncT + DecT(+TED)	31.33
EncT + DecT(+WMT)	31.27
EncT + DecT(+TED) + JO	35.87
EncT + DecT(+WMT) + JO	35.35

Comparing the "EncT + DecT(TED)" with the "EncT + DecT", we found that using an additional monolingual corpus TED in the decoder training improves the performance of the Transformer. This finding also holds when we used an additional corpus WMT. The results in the fifth and the sixth rows show that joint optimization can further improve the performance of the "EncT + DecT(TED)" and the "EncT + DecT(WMT)".

6. Conclusions

The data scarcity problem occurs when we use the Transformer in low-resource machine translation tasks. This paper proposed a simple but very effective algorithm to deal with the mismatch problem between the big model capacity of the Transformer and the small training dataset available. The training algorithm uses the BERT as a constraint of the latent semantic space and trains the Transformer in three stages, including encoder training, decoder training, and joint optimization. Independently training the encoder and the decoder helps to alleviate data scarcity and enables the Transformer to converge well. Joint optimization is used to make the encoder and decoder fully match in the latent space. With the supervision of the BERT, we transferred the BERT knowledge into the Transformer and made this model converge very well. The experiments on six low-resource translation tasks demonstrate that the Transformer trained by our algorithm significantly outperforms the baselines, including Transformer (base), Transformer (big), Transformer-based Transfer learning, RelPos, DeepRepre, and Bert-fused NMT, which are trained end-to-end in previous works. Compared with the Transformer (base) trained end-to-end, the Transformer (based) trained with our algorithm obtains 4.2%, 2.5%, 6.4%, 4.6%, 3.3%, and 4.5% performance improvement in terms of BLEU on the six tasks, respectively. Compared to other the BERT-fused methods, a major advantage of our approach is that it improves the performance of the Transformer without changing its structure and increasing the number of parameters. The experiments also suggest that the BERT fine-tuning is good for the performance when training the Transformer with our algorithm. A byproduct of the proposed algorithm is that it enables us to further improve the translation performance by training the decoder with the large-scale monolingual corpus of the target language. In summary, our approach significantly improves the performance of the Transformer in low-resource translation tasks. More details can be found in the supplementary document and code.

In the future, we will investigate the generalization of the proposed approach to other end-to-end NMT models.

Author Contributions: Funding acquisition, G.G.; Investigation, J.L.; Methodology, X.S.; Project administration, R.Y.; Resources, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by National Natural Science Foundation of China (Grant No. 61762069), Key Technology Research Program of Inner Mongolia Autonomous Region (Grant No. 2021GG0165), Key R&D and Achievement Transformation Program of Inner Mongolia Autonomous Region (Grant No. 2022YFHH0077), Big Data Lab of Inner Mongolia Discipline Inspection and Supervision Committee (Grant No. 21500-5206043).

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

References

1. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
2. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1724–1734.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
4. Luong, T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Lisbon, Portugal, 2015; pp. 1412–1421.
5. Sun, H.; Wang, R.; Chen, K.; Utiyama, M.; Sumita, E.; Zhao, T. Unsupervised Bilingual Word Embedding Agreement for Unsupervised Neural Machine Translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1235–1245.
6. Britz, D.; Goldie, A.; Luong, M.T.; Le, Q. Massive Exploration of Neural Machine Translation Architectures. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 1442–1451.
7. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1243–1252.
8. Ramesh, S.H.; Sankaranarayanan, K.P. Neural Machine Translation for Low Resource Languages using Bilingual Lexicon Induced from Comparable Corpora. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, New Orleans, LA, USA, 2–4 June 2018; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 112–119.
9. Lignos, C.; Cohen, D.; Lien, Y.C.; Mehta, P.; Croft, W.B.; Miller, S. The Challenges of Optimizing Machine Translation for Low Resource Cross-Language Information Retrieval. 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), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 3497–3502.
10. Nguyen, T.Q.; Chiang, D. Transfer Learning across Low-Resource, Related Languages for Neural Machine Translation. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Taipei, Taiwan, 27 November–1 December 2017; Asian Federation of Natural Language Processing: Taipei, Taiwan, 2017; pp. 296–301.
11. Kim, Y.; Gao, Y.; Ney, H. Effective Cross-lingual Transfer of Neural Machine Translation Models without Shared Vocabularies. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1246–1257.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
13. Pan, X.; Wang, M.; Wu, L.; Li, L. Contrastive Learning for Many-to-many Multilingual Neural Machine Translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual Event, 1–6 August 2021; pp. 244–258.
14. Sennrich, R.; Haddow, B.; Birch, A. Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 86–96. [CrossRef]
15. Baldi, P.; Vershynin, R. The capacity of feedforward neural networks. *Neural Netw.* **2019**, *116*, 288–311. [CrossRef] [PubMed]
16. Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; Liu, T. Incorporating BERT into Neural Machine Translation. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
17. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 5926–5936.
18. Conneau, A.; Lample, G. Cross-lingual language model pretraining. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.

19. Clinchant, S.; Jung, K.W.; Nikoulina, V. On the use of BERT for Neural Machine Translation. In Proceedings of the 3rd Workshop on Neural Generation and Translation, Hong Kong, China, 4 November 2019; pp. 108–117.
20. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3104–3112.
21. Weng, R.; Wei, H.; Huang, S.; Yu, H.; Bing, L.; Luo, W.; Chen, J. Gret: Global representation enhanced transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 9258–9265.
22. Bapna, A.; Chen, M.; Firat, O.; Cao, Y.; Wu, Y. Training Deeper Neural Machine Translation Models with Transparent Attention. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 3028–3033.
23. Wang, Q.; Li, B.; Xiao, T.; Zhu, J.; Li, C.; Wong, D.F.; Chao, L.S. Learning Deep Transformer Models for Machine Translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1810–1822.
24. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the NAACL-HLT, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
25. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5753–5763.
26. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
27. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
28. Yang, J.; Wang, M.; Zhou, H.; Zhao, C.; Zhang, W.; Yu, Y.; Li, L. Towards making the most of bert in neural machine translation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 9378–9385.
29. Zoph, B.; Yuret, D.; May, J.; Knight, K. Transfer Learning for Low-Resource Neural Machine Translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1568–1575.
30. Ahmadnia, B.; Serrano, J.; Haffari, G. Persian-Spanish Low-Resource Statistical Machine Translation Through English as Pivot Language. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP, Varna, Bulgaria, 2–8 September 2017; INCOMA Ltd.: Varna, Bulgaria, 2017; pp. 24–30.
31. He, D.; Xia, Y.; Qin, T.; Wang, L.; Yu, N.; Liu, T.Y.; Ma, W.Y. Dual learning for machine translation. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; Volume 29.
32. Ahmadnia, B.; Dorr, B.J. Augmenting neural machine translation through round-trip training approach. *Open Comput. Sci.* **2019**, *9*, 268–278. [CrossRef]
33. Xu, T.; Ozbek, O.I.; Marks, S.; Korrapati, S.; Ahmadnia, B. Spanish-Turkish Low-Resource Machine Translation: Unsupervised Learning vs Round-Tripping. *Am. J. Artif. Intell.* **2020**, *4*, 42–49. [CrossRef]
34. Chronopoulou, A.; Stojanovski, D.; Fraser, A. Improving the lexical ability of pretrained language models for unsupervised neural machine translation. *arXiv* **2021**, arXiv:2103.10531.
35. Atrio, A.R.; Popescu-Belis, A. On the Interaction of Regularization Factors in Low-resource Neural Machine Translation. In Proceedings of the 23rd Annual Conference of the European Association for Machine Translation, Ghent, Belgium, 1–3 June 2022; European Association for Machine Translation: Ghent, Belgium, 2022; pp. 111–120.
36. Qi, Y.; Sachan, D.; Felix, M.; Padmanabhan, S.; Neubig, G. When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation? In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 529–535.
37. Wang, Y.; Zhai, C.; Awadalla, H.H. Multi-task Learning for Multilingual Neural Machine Translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 1022–1034.
38. Tang, Y.; Tran, C.; Li, X.; Chen, P.J.; Goyal, N.; Chaudhary, V.; Gu, J.; Fan, A. Multilingual translation from denoising pre-training. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 3450–3466.
39. Chi, Z.; Dong, L.; Ma, S.; Huang, S.; Singhal, S.; Mao, X.L.; Huang, H.Y.; Song, X.; Wei, F. mT6: Multilingual Pretrained Text-to-Text Transformer with Translation Pairs. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021; pp. 1671–1683.
40. Rothe, S.; Narayan, S.; Severyn, A. Leveraging pre-trained checkpoints for sequence generation tasks. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 264–280. [CrossRef]
41. 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, 6–12 July 2002; pp. 311–318.
42. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv* **2019**, arXiv:abs/1910.03771.
43. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
44. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. *Comput. Sci.* **2015**. [CrossRef]

45. Dou, Z.Y.; Tu, Z.; Wang, X.; Shi, S.; Zhang, T. Exploiting Deep Representations for Neural Machine Translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 4253–4262.
46. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-Attention with Relative Position Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 464–468.