electronics

MDPI

Article

Neural Machine Translation of Electrical Engineering Based on Integrated Convolutional Neural Networks

Zikang Liu [1,2], Yuan Chen [3] and Juwei Zhang [1,2,*]

[1] School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China; zikangliu2023@163.com
[2] Henan Province New Energy Vehicle Power Electronics and Power Transmission Engineering Research Center, Luoyang 471023, China
[3] School of Foreign Languages, Henan University of Science and Technology, Luoyang 471023, China; 9903671@haust.edu.cn
* Correspondence: juweizhang@haust.edu.cn

Abstract: Research has shown that neural machine translation performs poorly on low-resource and specific domain parallel corpora. In this paper, we focus on the problem of neural machine translation in the field of electrical engineering. To address the mistranslation caused by the Transformer model's limited ability to extract feature information from certain sentences, we propose two new models that integrate a convolutional neural network as a feature extraction layer into the Transformer model. The feature information extracted by the CNN is fused separately in the source-side and target-side models, which enhances the Transformer model's ability to extract feature information, optimizes model performance, and improves translation quality. On the dataset of the field of electrical engineering, the proposed source-side and target-side models improved BLEU scores by 1.63 and 1.12 percentage points, respectively, compared to the baseline model. In addition, the two models proposed in this paper can learn rich semantic knowledge without relying on auxiliary knowledge such as part-of-speech tagging and named entity recognition, which saves a certain amount of human resources and time costs.

Keywords: neural machine translation; feature information; convolutional neural network; electrical engineering; low resource

Citation: Liu, Z.; Chen, Y.; Zhang, J. Neural Machine Translation of Electrical Engineering Based on Integrated Convolutional Neural Networks. *Electronics* **2023**, *12*, 3604. https://doi.org/10.3390/electronics12173604

Academic Editor: Arkaitz Zubiaga

Received: 3 August 2023
Revised: 17 August 2023
Accepted: 24 August 2023
Published: 25 August 2023

1. Introduction

Neural machine translation (NMT) aims to use computers to translate one language into another, and it plays a critical role in various scientific fields. Since 2014, NMT has developed rapidly, from recursive neural networks [1], to convolutional neural networks [2], and then to the Transformer neural network based on self-attention [3], which has achieved good results. Among several NMT models, Transformer performs the best, both in terms of efficiency and translation quality.

As various scientific fields continue to develop, the demand for NMT is also increasing rapidly. Different professional fields have different professional corpora, and some fields have very limited parallel corpora resources. Traditional NMT cannot meet the translation needs of some professional fields. The English–Chinese corpus in the field of electrical engineering is a typical low-resource corpus. The traditional Transformer does not perform well in the English–Chinese corpus in the field of electrical engineering, often causing mistranslation or misinterpretation of certain feature information in sentences, which makes it difficult for personnel in the electrical industry to use professional equipment and read professional English literature. The field of electrical engineering plays a crucial role in the development of many scientific fields. Therefore, it is essential to study how to design an efficient and stable model on a small-scale parallel corpus to improve the current situation of NMT in the field of electrical engineering.

Low-resource neural machine translation has long been an area of interest in natural language processing, and many researchers have made significant efforts to address this problem. Common improvement methods include data augmentation, introducing prior knowledge, and structural improvements. Tonja used monolingual source-side data to improve low-resource neural machine translation and achieved significant results on the Wolaytta–English corpus, further fine-tuning the best-performing self-learning model which resulted in +1.2 and +0.6 BLEU score improvements for Wolaytta–English and English–Wolaytta translations, respectively [4]. Mahsuli, MM proposed a method to model the length of a target (translated) sentence given the source sentence using a deep recurrent neural structure—and apply it to the decoder side of neural machine translation systems to generate translation sentences with appropriate lengths which have a better quality [5]. Pham, NL; Nguyen, V; and Pham, TV used back-translation to enhance the parallel database of English–Vietnamese machine translation, significantly improving the translation quality of the model [6]. Laskar, SR improved English–Assamese machine translation through pre-training models, and the best MNMT model, Transformer (transliteration-based phrase-augmentation), attained scores of +0.58, +1.86 (BLEU) [7]. Park, YH enhanced low-resource neural machine translation data through EvalNet and the NMT systems for English–Korean and English–Myanmar, built with the guidance of EvalNet, and achieved 0.1~0.9 gains in BLEU scores [8]. While these methods have achieved good results, they often require significant time and cost in the data preprocessing stage and have certain drawbacks. Dhar, P introduced bilingual dictionaries to improve Sinhala–English, Tamil–English, and Sinhala–Tamil translation and introduced a weighted mechanism based on small-scale bilingual dictionaries to improve the measurement of semantic similarity between sentences and documents [9]. Gong, LC achieved good results on several low-resource datasets by guiding self-attention with syntactic graphs [10]. Hlaing, ZZ added an additional encoder to the transformer model to introduce part-of-speech tagging, improving Thai-to-Myanmar, Myanmar-to-English, and Thai-to-English translation, outperforming such models developed through the existing Thai POS tagger in terms of BLEU scores (+0.13) and chrF scores (+0.47) for Thai-to-Myanmar, and BLEU scores (+0.75) and chrF scores (+0.72) for Myanmar-to-Thai translation pairs [11]. Considering that convolutional neural networks can extract feature information from sentences, this paper integrates a convolutional neural network as a feature extraction layer into the Transformer model. This method can introduce feature information into the Transformer model without additional processing of the corpus, improving the translation quality of the Transformer model while also saving research time and costs. The main contributions of this article are as follows:

In order to address the issue of feature information misinterpretation and omission in the corpus of electrical engineering with Transformer, a method is proposed to integrate a convolutional neural network as a feature extraction layer into Transformer, which effectively improves the translation accuracy of the model.

Two new model structures are proposed based on Transformer, and the specific structures of the two models are introduced in Sections 3 and 4, respectively.

Comparative experiments and ablation experiments are designed to verify the performance of the two models proposed in this paper on the dataset of electrical engineering, and their performance is compared with the baseline model, demonstrating that the Transformer model integrated with a convolutional neural network has a better performance.

2. Model

This paper adopts the Transformer model proposed by Google's machine translation team in 2017 as the baseline model, which mainly consists of four parts: input layer, output layer, encoder, and decoder (the structure of the baseline model [3] is shown in Figure 1). Since its introduction, Transformer has shown an outstanding performance in many natural language-processing tasks. The models proposed in this paper are all based on the baseline model but have improved upon it. Considering the weak ability of Transformer

to extract local feature information, a convolutional neural network can be used to extract feature information, which can compensate for the shortcomings of Transformer. In this paper, we improve the traditional Transformer structure by integrating a convolutional neural network composed of pooling and convolutional layers as a feature information extraction layer into Transformer. We fuse the feature information separately at the source language side and target language side and obtain two new models: Source-Side-CNN-Transformer (SSCT) and Target-Side-CNN-Transformer (TSCT).

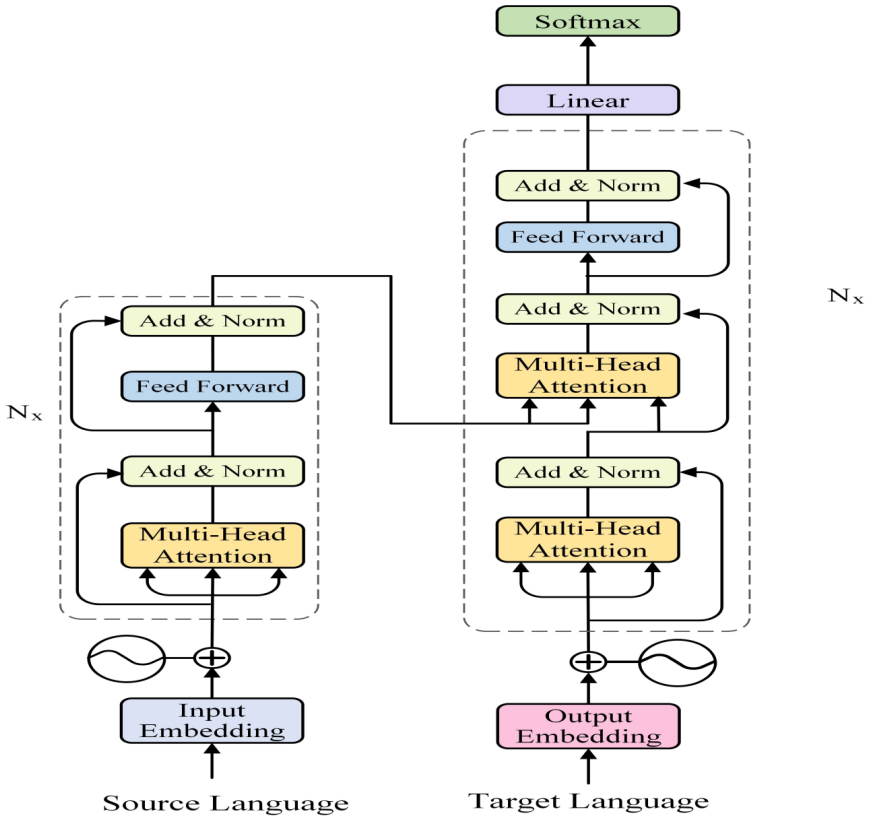

Figure 1. Baseline model.

The structure of SSCT is shown in Figure 2. Compared to the baseline model, SSCT adds a feature information extraction layer on the left side of the encoder, which consists of convolutional and pooling layers. Its purpose is to perform local feature extraction on the source language vectors after passing through the embedding layer. In order to integrate the convolutional neural network as the feature extraction layer into the Transformer model, SSCT connects the convolutional layer with the embedding layer of the baseline model, facilitating the convolutional layer to extract features from the source language vectors. In addition, a multi-head attention mechanism is added to the entire model framework, allowing the pooling layer of the feature extraction layer to be associated with the encoder part of the baseline model, thereby fully integrating the feature extraction layer into the Transformer model. The role of the multi-head attention mechanism between the convolutional neural network and the source language encoder is to fuse the locally extracted feature vectors from the feature extraction layer with the output vectors from the encoder. The fused vector is then used as the input to the decoder's context multi-head attention mechanism, which is associated with the contextual information in the decoder. This enables the decoder to effectively learn the relationship between global information and feature information.

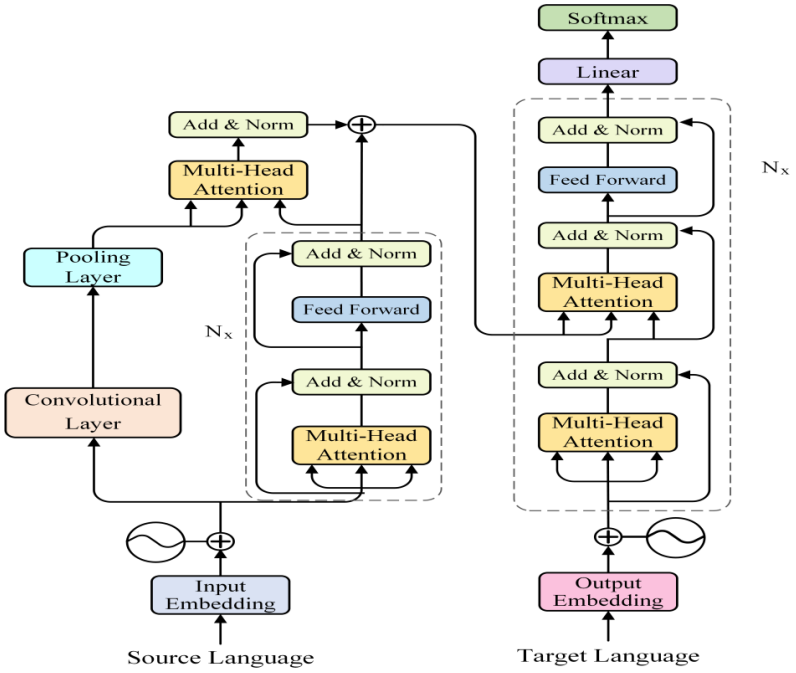

Figure 2. Source-Side-CNN-Transformer.

TSCT and SSCT share the same structure for the feature information extraction layer but differ in their integration methods. As shown in Figure 3, TSCT also connects the convolutional layer to the embedding layer. However, in the second sub-layer of the decoder, an additional multi-head attention mechanism is introduced (on the left side of the global attention mechanism). This allows the pooling layer of the feature extraction layer to be connected to the global attention of the decoder, enabling it to receive feature information. The attention calculation is performed between the feature information and the internal information of the decoder, allowing the decoder to learn the relationship between feature information and global information comprehensively. This integration reduces translation errors, mistranslations, and other phenomena that occur in the Transformer model.

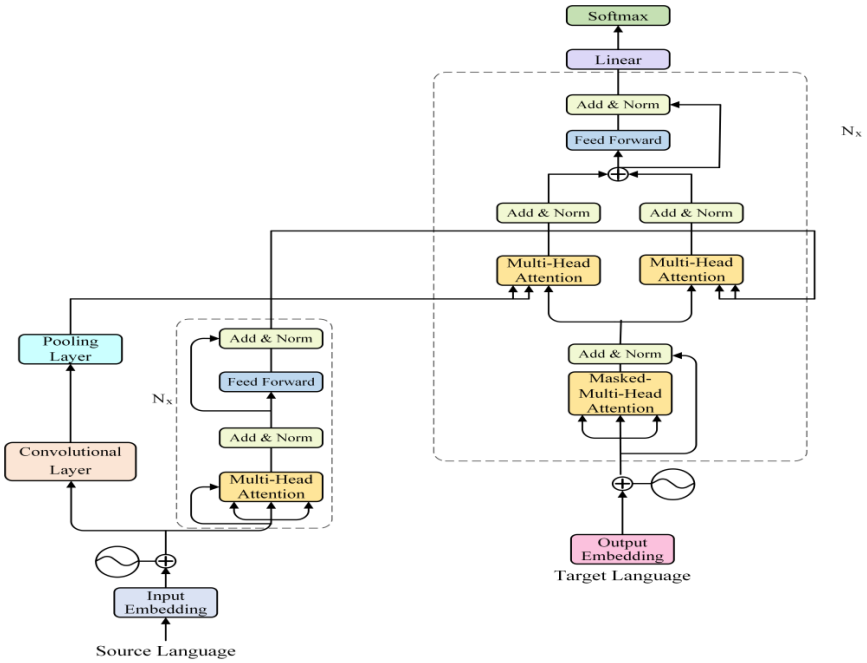

Figure 3. Target-Side-CNN-Transformer.

3. Source-Side-CNN-Transformer

3.1. Embedding and Positional Encoding

The Transformer model cannot directly process text sequences; thus, it needs to map raw data with high dimensions to low-dimensional data through embedding layers, converting text into vector form so that the data can be processed by neural networks [12,13]. For any language, the position and arrangement of words in a sentence are very important. They are not only a part of the grammatical structure of the sentence but also important concepts for expressing semantics. If the position and sequence of a word in a sentence are different, the meaning of the entire sentence will deviate. The Transformer model itself does not have the ability to learn sequential information like an RNN, so it needs a positional encoding layer to combine the sequential information with the word vectors and input them to the transformer, enabling the model to learn sequential information.

The two models proposed in this article have not made any changes to the embedding layer or positional encoding layer and still use the baseline model's embedding layer and positional encoding layer [3]. Defining the source language sequence after word segmentation as $X = [x_1, x_2, x_3, \ldots, x_n]$, the target language sequence is defined as $T = [t_1, t_2, t_3, \ldots, t_n]$, and the numerical identifier is defined as $I = [i_1, i_2, i_3, \ldots, i_n]$. Taking the embedding process of the source language as an example (the embedding process of the source language and the target language are the same), after linear transformation (Equation (1)), it is represented as E_x, and then the position information of each word (Equation (2)) is added to the embedding layer to obtain a result with positional information (Equation (3)):

$$E_x = W(I) \tag{1}$$

$$\begin{aligned} P_i(pos, 2i) &= \sin(pos/10000^{2i/\dim}) \\ P_i(pos, 2i+1) &= \cos(pos/10000^{2i/\dim}) \end{aligned} \tag{2}$$

$$E'_x = P_i + E_x \tag{3}$$

where I is the numerical identifier, $W()$ represents the linear transformation, E_x is the result after linear transformation, dim is the word vector dimension, P_i is the positional information, and E'_x is the word vector with positional information.

3.2. Encoder

In this section, no changes have been made to the encoder of the baseline model, which is composed of N = 6 independent layers stacked together. Each encoder contains two sublayers: multi-head self-attention mechanism and a feedforward neural network. Each sublayer is followed by a residual network and a normalization layer. The encoder takes the source language vectors processed by the embedding layer as the input for the multi-head self-attention mechanism and performs attention calculation and normalization processing on it (Equation (4)). Finally, the output of the encoder is obtained through the feedforward neural network (Equation (5)).

$$S_{self-attention} = Multihead(S^{i-1}, S^{i-1}, S^{i-1}) \tag{4}$$

$$S_{out} = Addnorm(FNN(S_{self-attention})) \tag{5}$$

S^{i-1} represents the output of the i-th layer of the encoder. The output of each layer is used as the input for the next layer. The input for the first layer of the encoder is the embedded source language vector. $Multihead()$ represents multi-head self-attention mechanism, $FNN()$ represents the feedforward neural network, and $Addnorm()$ represents the residual connection and layer normalization.

3.3. Feature Extraction Layer

Convolutional neural networks are generally used for image classification and object detection in the field of computer vision. However, since the proposal of Text CNN in

2014 [14], there have been more and more works applying convolutional neural networks to natural language processing tasks. Considering the strong ability of convolutional neural networks to extract local features, which can compensate for the weak ability of Transformer to extract local information, this article selects a convolutional neural network composed of a convolutional layer with a height of 3 and a width of 512 and a max-pooling layer with a height of 2 and a width of 1 as the feature extraction layer of the model (Figure 4). The parameters of the convolution kernel is shown in Table 1. The role of the convolutional layer is to extract feature information from the sentence, while the role of the max-pooling layer is to select the extracted feature information from the convolutional layer.

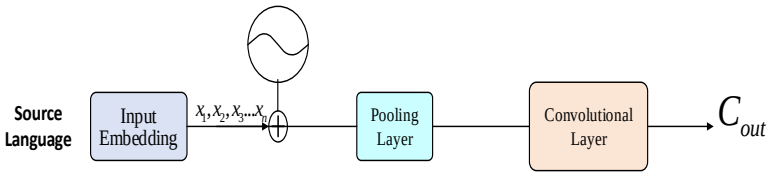

Figure 4. Feature information extraction layer.

Table 1. The parameters of the convolution kernel.

Parameters	
CNN_height	3
CNN_width	512
Input_channel	1
Output_channel	1
Pooling	Max_pooling
Pooling_height	2
Pooling_width	1
Act_function	ReLU
Num_fliter	51,200

3.3.1. Convolutional Layer

The input of the convolutional layer is the sentence of L which is represented as the $L * \text{dim}$ vector matrix M. The convolution is composed of $L * \text{dim}$ filters, each of which extracts features from the corresponding vector matrix, producing feature B_d^m (where the maximum sentence length L is 100 and dim is 512):

$$B_d^m = \text{ReLU}(f_m * M_{d:2*\text{dim}}) \tag{6}$$

where M represents the vector matrix, ReLU is the activation function, f_m represents the filters and L represents the maximum sentence length. Each filter extracts features from windows of each input matrix, producing feature vectors $B_m = \{B_1^m, B_2^m, B_3^m, \ldots, B_{\text{dim}}^m)$, and then $L * \text{dim}$ filters are used to process m sequentially, producing feature map C:

$$C = \{C_1, C_2, C_3, \ldots C_{2*\text{dim}}\} \tag{7}$$

where B_m represents the feature vectors and C represents the feature map.

3.3.2. Pooling Layer

The pooling layer first selects the maximum value in the adjacent feature maps p_m, and then normalizes the selected feature maps P using the tanh function to obtain the feature information C_{out}:

$$p_m = Max\{C_{2*\text{dim}-1}, C_{2*\text{dim}}\} \tag{8}$$

$$P = \{P_1, P_2, P_3, \ldots, P_{\text{dim}}\} \tag{9}$$

$$C_{out} = \{C_1, C_2, C_3, \ldots C_{2*\text{dim}}\} \tag{10}$$

The reason for selecting the max-pooling layer is because we believe that its mechanism can extract more precise feature information and reduce the impact of useless information. If we were to use the average pooling layer, the extracted information may contain more useless information, leading to the poor performance of the model. To validate our idea, we conducted a comparative experiment between models with a max-pooling layer and an average pooling layer. The experimental results (Table 2) demonstrated that the model with the max-pooling layer had a better performance. The relevant parameter settings and the models used for the comparative experiments on pooling layers are described in Sections 5.1 and 5.2 of the manuscript.

Table 2. Comparative experiment of pooling layer.

Model	BLEU/%	σ
SSCT-Max-Pooling	35.88	–
SSCT-Avg-Pooling	35.35	−0.53
TSCT-Max-pooling	35.37	–
TSCT-Avg-Pooling	34.86	−0.51

3.4. Attention Fusion Layer

In this article, the context multi-head attention mechanism is used to fuse all of the outputs of the encoder and the output of the feature extraction layer, which allows for sufficient correlation between the local feature information extracted by the convolutional neural network and the vectors output by the encoder. The calculation process of the fusion is shown in Equation (11).

$$F_{context-attention} = Addnorm(MultiHead(S_{out}, C_{out}, C_{out})) \tag{11}$$

F_{out}, S_{out}, and C_{out} represent the output of the attention fusion layer, encoder, and feature extraction layer, respectively. Influenced by previous works [15–17], S_{out} and $F_{context-attention}$ are concatenated along the last dimension of S_{out} to calculate the balancing factor:

$$y = Sigmoid([S_{out} : F_{context-attention}], \dim = -1) \tag{12}$$

where y is the balancing factor, $Sigmoid$ is the activation function, and the value of y is (0~1). Finally, a simple weight-based sum operation is adopted for the calculation of the attention fusion layer output:

$$F_{out} = y * S_{out} + (1 - y) * F_{context-attention} \tag{13}$$

3.5. Decoder

The decoder is similar to the encoder, consisting of N = 6 independent layers. However, each layer of the decoder has an additional sub-layer, which is composed of a multi-head attention mechanism, residual connections, and normalization. This sub-layer is used to receive outputs from the encoder. The calculation process of multi-head attention in the first sub-layer of the decoder is shown in Equation (14):

$$T_{self-decoder} = Addnorm(Multihead(T^{i-1}, T^{i-1}, T^{i-1})) \tag{14}$$

T^{i-1} represents the output of the i-th layer of the decoder. Each layer of the decoder uses the output of the previous decoder layer as its input. The input of the first decoder layer is the target language after embedding. In the second sub-layer, $T_{self-decoder}$ and F_{out} are used as inputs for the context multi-head attention calculation (Equation (15)). After

that, the output is passed through a feedforward neural network to obtain the final output of the decoder (Equation (16)).

$$T_{context-attention} = Addnorm(Multihead(T_{self-decoder}, F_{out}, F_{out})) \quad (15)$$

$$T_{out} = Addnorm(FNN(T_{context-attention})) \quad (16)$$

4. Target-Side-CNN-Transformer

Due to the fact that the embedding layer and feature extraction layer of TSCT are the same as those of SSCT, this chapter will not introduce these two parts in detail. Readers can refer to Sections 3.1 and 3.3 of this paper for specific details.

4.1. Encoder

The encoder used in the TSCT model is consistent with the encoders used in the baseline model and SSCT. Therefore, its structure will not be described in detail in this section (please refer to Section 3.2 for specific details on the calculation process of the TSCT encoder). The calculation process of the TSCT encoder is as follows.

$$S_{self-attention} = Multihead(S^{i-1}, S^{i-1}, S^{i-1}) \quad (17)$$

$$S_{out} = Addnorm(FNN(S_{self-attention})) \quad (18)$$

S^{i-1} represents the output of the i-th layer of the encoder. The output of each layer is used as the input for the next layer. The input for the first layer of the encoder is the embedded source language vector, and S_{out} represents the total output of the source language encoder.

4.2. Decoder

Since the feature information is the input to the decoder in the TSCT model, a CNN-Decoder attention mechanism needs to be added to the decoder unit of the baseline model to receive the output from the feature information extraction layer, facilitating further fusion of the feature information and global information. The improved decoder unit has three sub-layers: a self-attention sub-layer, a multi-head attention sub-layer composed of Encoder–Decoder and CNN-Decoder, and a fully connected feedforward network sub-layer.

The specific structure of the decoder is shown in Figure 5. The attention mechanisms in the TSCT model are Self-Decoder, Encoder–Decoder, and CNN-Decoder. The internal structures and calculation methods of these three attention mechanisms are the same, and their main difference lies in the query vector Q and key–value pairs K and V:

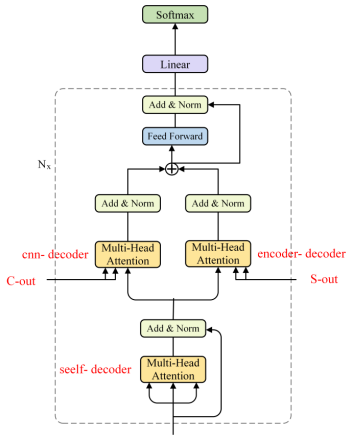

Figure 5. The decoder of TSCT.

The role of Self-Decoder is to learn the target information fully; thus, its query vector Q, key K, and value V are all source sentence vectors after embedding. The calculation process is shown in Equation (19).

$$T_{self-decoder} = Addnorm(Multihead(T^{i-1}, T^{i-1}, T^{i-1})) \tag{19}$$

Encoder–Decoder is mainly used to transmit encoder information to the decoder, so that better weight allocation can be achieved based on the target information for the source language representation vector obtained by the encoder during the decoding process. Therefore, the query vector Q of Encoder–Decoder is the output of Self-Decoder, and the key–value pairs K and V still come from the output vectors of the encoder and have the same numerical values. The calculation process is shown in the Equation (20).

$$T_{encoder-decoder} = Addnorm(Multihead(T_{self-decoder}, S_{out}, S_{out})) \tag{20}$$

The main role of CNN-Decoder is to transmit feature information to the decoder, so that the target information in the decoder can be fully associated with the feature information. Therefore, the query vector Q of CNN-Decoder is the output of Self-Decoder, and the K and V are the outputs of the feature information extraction layer.

$$T_{cnn-decoder} = Addnorm(Multihead(T_{self-decoder}, C_{out}, C_{out})) \tag{21}$$

By using a balancing mechanism to control the information flow [15–17], more valuable information can be obtained. In this paper, $T_{encoder-decoder}$ and $T_{cnn-decoer}$ are concatenated on the last dimension of $T_{encoder-decoder}$ to calculate the balancing coefficient y.

$$y = Sigmoid([T_{encoder-decoder} : T_{cnn-decoer}], \dim = -1) \tag{22}$$

A simple summation operation is performed to obtain T_{out}:

$$T_{out} = y * T_{encoder-decoder} + (1 - y) * T_{cnn-decoer} \tag{23}$$

5. Experiment

In this part, we conducted experiments and research about the two models proposed in this paper on the Chinese–English parallel corpus in the field of electrical engineering. We compared the models proposed in this paper with the baseline model and conducted ablation experiments on the two models proposed in this paper.

5.1. Dataset

All the datasets used in this paper are Chinese–English parallel corpora in the field of electrical engineering, mainly collected from certain Chinese and English materials in the field of electrical engineering, including several professional books [18–21], equipment manuals, literature, and some technical forums and official websites related to electrical engineering. The training set used in the experiment has about 190,000 bilingual parallel corpora, and the validation set and test set each have 2000 bilingual parallel sentence pairs.

5.2. Parameter Settings

We used the open-source system OpenNMT [22] to implement the baseline model Transformer. In terms of data processing, the sentence length in the corpus was limited to within 100, that is, sentences longer than 100 were filtered out, and the vocabulary size was set to 44,000. Chinese segmentation was conducted using Jieba, and English segmentation was conducted using NLTK. During the training process, the dimension of word vectors and the hidden layer dimension of the encoder and decoder were both set to 512. The batch size was set to 64, and the Adam optimization algorithm was used. The dropout rate was set to 0.1. A total of 25,000 steps were trained in this experiment, and the model was validated every 1000 steps. The beam search method was used in decoding, with a beam

size of 5 and other parameters using OpenNMT's default parameters. All parameters in the experiment were kept consistent (Table 3), and the translation results were evaluated using BLEU [23]. All experiments were conducted on the same GPU device, specifically the RTX-3090 model.

Table 3. Experimental parameters.

Parameters	
Baseline	Transformer
Word_max_length	100
Hidden_size	512
Word_vec_size	512
dropout	0.1
Optimizer	Adam
Learning_rate	2
Label_smoothing	0.1
Beam_size	5
Enc_layer	6
Dec_layer	6
Transformer_ff	2048
Src_vocab_size	44,000
Tgt_vocab_size	44,000
Batch_size	64
Train_steps	25,000
Vaild_steps	1000
Report_every	100
seed	1234
adam_beta2	0.998

5.3. Experiment and Analysis

5.3.1. Source-Side Model Experiment

In this section, we conducted comparative experiments between the Source-Side-CNN-Transformer and the baseline model. The Source-Side-CNN-Transformer integrates the output of the feature extraction layer with the output of the encoder through a multi-head attention mechanism, and then gates the fused vector with the residual-connected encoder output. The obtained vector is used as the input of the decoder's multi-head attention mechanism and associated with the target information. The experimental results of this method are shown in Table 4, and the improved model has a 1.63% higher BLEU score than the baseline model.

Table 4. Source-side experiment.

Model	**BLEU/%**	σ
Baseline	34.25	–
SSCT	35.88	+1.63

This experiment shows that after the fusion of vectors with local feature information is completed at the source language end, it can effectively improve the translation performance of Transformer and mitigate its weakness in extracting local feature information.

5.3.2. Target-Side Model Experiment

In this section, we conducted comparative experiments between TSCT and the baseline model. This model integrates the output of the feature extraction layer into the target end of Transformer, enabling it to learn the feature information (Table 5).

Table 5. Target-side experiment.

Model	BLEU/%	σ
Baseline	34.25	–
TSCT	35.37	+1.12

The experimental results show that fusing local information with global information in the target end can improve the performance of Transformer.

5.3.3. Ablation Experiment

To prove that the Transformer integrated with convolutional neural network has a better performance, this section conducted comparative experiments between the original models proposed in Sections 3 and 4 and the models with the convolutional neural network removed (SST and TST represent the models after removing the feature extraction layer from SSCT and TSCT, respectively). The experimental results are as shown in Tables 6 and 7.

Table 6. SSCT-ablation experiment.

Model	BLEU/%	σ
SSCT	35.88	–
SST	34.79	−1.09

Table 7. TSCT-ablation experiment.

Model	BLEU/%	σ
TSCT	35.37	–
TST	34.50	−0.87

5.3.4. Comparison Experiment

In order to further demonstrate the effectiveness of our approach, comparative experiments were conducted on an electrical engineering dataset with our model and baseline models such as Sentence-level [24], Key Information Fusion [25], Pos Fusion [11], and Prior Knowledge [26]. The experimental conditions were kept consistent, and the results are shown in Table 8.

Table 8. The BLEU values of comparison experiments.

Model	BLEU/%	Time/min	σ
Baseline	34.25	141	–
Sentence-level	34.93	203	+0.68
Key Information Fusion	34.97	479	+0.72
Pos Fusion	35.27	471	+1.02
SSCT	35.88	179	+1.63
Prior Knowledge	34.82	486	+0.57

Sentence-level: Proposed in 2019 by Kehai Chen et al., this method uses a convolutional neural network to extract sentence-level contextual information and integrates it into the Transformer model to improve translation performance.

Key Information Fusion: Proposed in 2023 by Shije Hu et al., this method utilizes a dual-encoder structure to incorporate key information from the text into the Transformer model, aiming to enhance its performance.

Pos Fusion: Proposed in 2022 by Z et al., this method first performs part-of-speech tagging on the corpus, and then integrates the part-of-speech tagging information into the

Transformer model using a dual-encoder structure. A stacked decoder structure is used to associate the part-of-speech tagging information with the target information.

Prior Knowledge: Proposed in 2022 by Rui Wang et al., this method extracts prior knowledge which is then fused with the source language information in the form of matrix-vector. This enriches the semantic knowledge learned by the Transformer model. From the comparative experiment results, it can be observed that our approach achieves a better translation performance compared to other models on the electrical engineering dataset.

Compared to the baseline model, our approach shows noticeable improvements in BLEU. Additionally, our approach outperforms the other comparative models in all the metrics, indicating that it effectively utilizes contextual information and key information to enhance translation quality.

The baseline, sentence-level, and SSCT models do not integrate prior knowledge into the Transformer model, and the processing time or extraction time consumed is zero. The Key Information Fusion, Prior Knowledge, and Pos Fusion models all need to extract and process prior knowledge. The tools used and the time spent in each stage are shown in Table 9.

$$Time = time_{execution} + time_{extraction} + time_{process\,sing} \tag{24}$$

where $Time$ represents the total time, $time_{execution}$ represents the execution time, and $time_{process\,sing}$ and $time_{extraction}$ represent the processing time that the model needs to process or extract prior knowledge.

Table 9. Time cost.

Method	tool	Extraction Time/min	Processing Time/min	Execution Time/min
Key Information	YAKE	133	143	203
Pos Fusion	Stanford Tagger	121	157	193
Prior Knowledge	Stanford CoreNLP	139	158	189

5.3.5. Extended Experiment

To further investigate the universality of the proposed method, the models presented in this paper were tested on publicly available general corpora, and the experimental results are shown in Table 10.

Table 10. The result of the extended experiment.

Dataset	Model	BLEU/%	σ
WMT2017(CN-EN)	Baseline	22.82	–
	SSCT	24.09	+1.27
	TSCT	23.57	+0.75

The experimental results indicate that even on general Chinese-to-English corpora, the two models proposed in this paper still exhibit a certain effectiveness. This also confirms the validity of the improvements made to the Transformer structure in this study. Integrating convolutional neural networks as the feature extraction layer into the Transformer's structure indeed enhances the translation capability of the model. With the improved model obtaining crucial local information, it continuously learns richer semantic relationships and correct logical connections during training, leading to more accurate translation outcomes.

5.3.6. Analysis

Based on the results of the experiments in Section 5.3, we draw the following conclusions:

The results of Sections 5.3.1 and 5.3.2show that the two new structures proposed in this paper based on Transformer significantly improve the performance compared to the baseline model, and fusing feature information with Transformer at either the source or target end can improve translation quality. As can be seen from the translation examples in Tables 11 and 12 (These red fonts in the table are technical terms), compared to the baseline model, SSCT and TSCT translate key information in the sentence more accurately, and the translated sentences are also closer to the reference translation.

Table 11. Translation samples (a).

Source text	A Configuration Scheme of Two Automatic Bus Transfer Equipment for 10 kV Sections in a Substation with Three Main Transformers
Reference	在一个有两台主变压器的变电站中，针对10千伏区段的两个自动母线转换设备的配置方案。
Translation of baseline model	
两台主变压器转换电站10 kV段2台自动母线传输设备配置方案/Configuration scheme of two main transformers converting power station with two automatic bus transmission equipment in the 10 kV section	
Translation of Source-Side-CNN-model	
为一个具有两台主变压器的变电站，设计10千伏范围的两个自动母线转换设备的装置方案。/Designing a device scheme for two automatic busbar transfer devices within the 10 kV range for a substation equipped with two main transformers.	
Translation of Target-Side-CNN-model	
在两台主变压器的变电站中，为10千伏区段的两个自动主线转换设备的设计的配置方案。/Configuration scheme for two automatic busbar transfer devices designed for the 10 kV section in a substation with three main transformers.	

Table 12. Translation samples (b).

Source text	The paper puts forward impact factor analysis of the reliability in bulk power system using power flow tracing method on the basic of power flow tracing load-shedding model.
Reference	本文在潮流跟踪负荷削减模型基础上，提出了大电力系统可靠性影响因素分析的潮流跟踪法。
Translation of baseline model	
在潮流跟踪负荷减少模型的基础上提出了电子系统的可靠性影响因素分析的潮流跟随法。/Based on the load reduction model of power flow tracking, a power flow following method for the analysis of reliability influencing factors ofelectronic systems is proposed.	
Translation of Source-side-CNN-model	
这篇文章在潮流跟踪负荷削减模型的削减基础上提出了适用于大电力系统可靠性影响因素分析的潮流跟踪法。/This article proposes a load flow tracking method for analyzing the reliability impact factors in large-scale power systems, based on the load reduction model.	
Translation of Target-side-CNN-model	
本文在削减模型的基础上提出了较大的电力系统可靠性影响因素分析的潮流跟踪法。/The present study introduces the load flow tracking method for analyzing the reliability impact factors in large-scale power systems, building upon an existing load reduction model.	

SSCT performs better than TSCT, and Transformer has better results in vector fusion at the source end. The reason for this result may be that Transformer loses more source language vectors when fusing vectors at the target end, causing the fused vector of $T_{encoder-decoder}$ and $T_{cnn-decoder}$ lose some semantic information, resulting in slightly lower performance for TSCT than SSCT.

The ablation experiment in Section 5.3.3 further proves that the method proposed in this paper is effective. Integrating the convolutional neural network as a feature extraction layer into Transformer can enable it to learn the feature information in the sentence and improve its translation performance.

The results of comparative experiments show that compared with the previous models, the model proposed in this paper has a better performance on datasets in the field of electrical engineering and saves a lot of time and cost than other models using prior knowledge.

The results of extended experiments prove that the method in this paper is also applicable to the general corpus, not only in the field of electrical engineering. The method proposed in this article has a certain versatility.

6. Conclusions

In this paper, we improved the Transformer model by integrating the convolutional neural network as a feature extraction layer into its overall structure and obtained two new models, SSCT and TSCT, which respectively perform vector fusion of feature information at the source and target sides. This allows the improved Transformer model to learn semantic knowledge containing feature information.

SSCT introduced a multi-head attention mechanism between the feature information extraction layer and the source language encoder, which is used to fuse the feature vector and encoder output vector. The fused vector is used as the K and V vector in the decoder multi-head attention mechanism and is calculated with the internal vector of the encoder layer. Through this method, the model can continuously learn semantic information containing feature vectors during training, thereby enhancing the performance of the translation model. TSCT has the same feature information extraction layer as SSCT, but the decoder unit of TSCT adds a multi-head attention mechanism, which uses the output of the feature information extraction layer as the Q vector and the output of the first sub-layer of the decoder as the K and V vectors. The result of the calculation is fused with the output of the second sub-layer of the decoder, allowing the Transformer model to further learn the relationship between feature information and global information. The results of the comparative experiments and ablation experiments designed in this paper show that the BLEU values of SSCT and TSCT are 1.63 and 1.12 percentage points higher than the baseline model, respectively, which fully demonstrates the effectiveness of our proposed method.

Indeed, while the model designed in this paper exhibits an excellent performance on the English-to-Chinese dataset in the electrical engineering domain, its performance may show some decline on large-scale parallel corpora, indicating the limitations of this approach. In future work, we plan to explore the integration of bidirectional gated recurrent units (GRUs) and convolutional neural networks (CNNs) into both the source and target sides of the Transformer model. By doing so, the improved Transformer model will be able to learn from both memory information and local information, enhancing the overall performance and generalizability of the model. This will enable the model to achieve better results on corpora from various domains.

Author Contributions: Research conceptualization and model building: Z.L.; data collection: Z.L. and Y.C.; experiment design: Z.L., Y.C. and J.Z.; manuscript preparation: Z.L.; manuscript review: Z.L., Y.C. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Partial data are available (https://github.com/zikangliu0612/zikang). (https://opennmt.net). The access date accessed on 23 August 2023.

Conflicts of Interest: The authors declare that they have no conflict of interest to report regarding the present study.

References

1. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
2. 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, 6–17 August 2017.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
4. Tonja, A.L.; Kolesnikova, O.; Gelbukh, A.; Sidorov, G. Low-Resource Neural Machine Translation Improvement Using Source-Side Monolingual Data. *Appl. Sci.* **2023**, *13*, 1201. [CrossRef]
5. Mahsuli, M.M.; Khadivi, S.; Homayounpour, M.M. LenM: Improving Low-Resource Neural Machine Translation Using Target Length Modeling. *Neural Process. Lett.* **2023**, 1–32. [CrossRef]
6. Pham, N.L.; Pham, T.V. A Data Augmentation Method for English-Vietnamese Neural Machine Translation. *IEEE Access* **2023**, *11*, 28034–28044.

7. Laskar, S.R.; Paul, B.; Dadure, P.; Manna, R.; Pakray, P.; Bandyopadhyay, S. English–Assamese neural machine translation using prior alignment and pre-trained language model. *Comput. Speech Lang.* **2023**, *82*, 101524. [CrossRef]
8. Park, Y.H.; Choi, Y.S.; Yun, S.; Kim, S.H.; Lee, K.J. Robust Data Augmentation for Neural Machine Translation through EVALNET. *Mathematics* **2022**, *11*, 123. [CrossRef]
9. Dhar, P.; Bisazza, A.; van Noord, G. Evaluating Pre-training Objectives for Low-Resource Translation into Morphologically Rich Languages. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; pp. 4933–4943.
10. Gong, L.; Li, Y.; Guo, J.; Yu, Z.; Gao, S. Enhancing low-resource neural machine translation with syntax-graph guided self-attention. *Knowl. Based Syst.* **2022**, *246*, 108615.
11. Hlaing, Z.Z.; Thu, Y.K.; Supnithi, T.; Netisopakul, P. Improving neural machine translation with POS-tag features for low-resource language pairs. *Heliyon* **2022**, *8*, e10375. [CrossRef] [PubMed]
12. Ghannay, S.; Favre, B.; Esteve, Y.; Camelin, N. Word embedding evaluation and combination. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 300–305.
13. Levy, O.; Goldberg, Y. Neural word embedding as implicit matrix factorization. *Adv. Neural Inform. Process. Syst.* **2014**, *27*, 1–9.
14. Zhang, Y.; Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv* **2015**, arXiv:1510.03820.
15. Gulcehre, C.; Firat, O.; Xu, K.; Cho, K.; Barrault, L.; Lin, H.C.; Bougares, F.; Schwenk, H.; Bengio, Y. On using monolingual corpora in neural machine translation. *arXiv* **2015**, arXiv:1503.03535.
16. Wang, Y.; Xia, Y.; Tian, F.; Gao, F.; Qin, T.; Zhai, C.X.; Liu, T.Y. Neural machine translation with soft prototype. *Adv. Neural Inform. Process. Syst.* **2019**, *32*, 1–10.
17. Cao, Q.; Xiong, D. Encoding gated translation memory into neural machine translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October 31–4 November 2018; pp. 3042–3047.
18. Bimal, K. *Modern Power Electronics and AC Drives*; Prentice-Hall: Hoboken, NJ, USA, 2001.
19. Bimal, K. *Modern Power Electronics and AC Drive*; Wang, C.; Zhao, J.; Yu, Q.; Cheng, H., Translators; Machinery Industry Press: Beijing, China, 2005.
20. Wang, Q.; Glover, J.D. *Power System Analysis and Design (Adapted in English)*; Machinery Industry Press: Beijing, China, 2009.
21. Glover, J.D. *Power System Analysis and Design (Chinese Edition)*; Wang, Q.; Huang, W.; Yan, Y.; Ma, Y., Translators; Machinery Industry Press: Beijing, China, 2015.
22. Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; Rush, A.M. Opennmt: Open-Source Toolkit for Neural Machine Translation. *arXiv* **2017**, arXiv:1701.02810.
23. 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 on Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
24. Chen, K.; Wang, R.; Utiyama, M.; Sumita, E.; Zhao, T. Neural machine translation with sentence-level topic context. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 1970–1984. [CrossRef]
25. Hu, S.; Li, X.; Bai, J.; Lei, H.; Qian, W.; Hu, S.; Zhang, C.; Kofi, A.S.; Qiu, Q.; Zhou, Y.; et al. Neural Machine Translation by Fusing Key Information of Text. *CMC Comput. Mater. Contin.* **2023**, *74*, 2803–2815. [CrossRef]
26. Chen, K.; Wang, R.; Utiyama, M.; Sumita, E. Integrating prior translation knowledge into neural machine translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *30*, 330–339. [CrossRef]