applied sciences

MDPI

Article

Neural Machine Translation of Electrical Engineering with Fusion of Memory Information

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

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

Abstract: This paper proposes a new neural machine translation model of electrical engineering that combines a transformer with gated recurrent unit (GRU) networks. By fusing global information and memory information, the model effectively improves the performance of low-resource neural machine translation. Unlike traditional transformers, our proposed model includes two different encoders: one is the global information encoder, which focuses on contextual information, and the other is the memory encoder, which is responsible for capturing recurrent memory information. The model with these two types of attention can encode both global and memory information and learn richer semantic knowledge. Because transformers require global attention calculation for each word position, the time and space complexity are both squared with the length of the source language sequence. When the length of the source language sequence becomes too long, the performance of the transformer will sharply decline. Therefore, we propose a memory information encoder based on the GRU to improve this drawback. The model proposed in this paper has a maximum improvement of 2.04 BLEU points over the baseline model in the field of electrical engineering with low resources.

Keywords: neural machine translation; memory information; gated recurrent unit; electrical engineering; low resource

Citation: Chen, Y.; Liu, Z.; Zhang, J. Neural Machine Translation of Electrical Engineering with Fusion of Memory Information. *Appl. Sci.* **2023**, *13*, 10279. https://doi.org/10.3390/app131810279

Academic Editor: Christos Bouras

Received: 13 August 2023
Revised: 30 August 2023
Accepted: 7 September 2023
Published: 13 September 2023

1. Introduction

With the rapid development of the information age, virtual connections across various fields around the world have been established through the internet [1]. The application of various scientific fields has gradually increased, and communication between different fields has become more frequent. The language barrier between different scientific fields has become a bottleneck for academic communication, which has greatly hindered the development of technology. Machine translation is a field in natural language processing, which is defined as the process of translating words, sentences, paragraphs, and entire texts from one language to another [2], and it plays a crucial role in the development of various scientific fields.

In 2013, Nal Kalchbrenner and Phil Blunsom proposed a new end-to-end encoder–decoder structure for machine translation, which gave rise to neural machine translation [3]. In the following years, recursive neural networks [4], convolutional neural networks [5], and transformers [6] were successively proposed, among which the transformers have the most comprehensive performance and have become the focus of many researchers in recent years.

Research has shown that transformers have achieved great success on large corpora, but their performance is relatively poor on language pairs with limited training data (also known as low resource) [7]. The main reason for this phenomenon is that models trained on general corpora cannot correctly translate specialized semantics in some specific fields,

and the same word often has vastly different meanings in different fields. Therefore, designing suitable neural machine translation models for specific fields is extremely necessary. This paper focuses on English-to-Chinese translation in the field of electrical engineering, which is a typical low-resource neural machine translation task. Many scientific fields require the application of knowledge related to electrical engineering, and designing a translation model that is adapted to the field of electrical engineering can facilitate academic communication between different countries and promote technological development. Considering that the traditional transformer may result in ambiguity when translating complex sentences with strong specialization, this paper proposes a memory information encoder to improve this situation.

The main contributions of this paper are as follows:

In order to optimize the performance of the transformer in translating complex sentences and compensate for its weakness in capturing long-term dependencies in language sequences, a new encoder structure is proposed by combining a GRU with an attention mechanism and integrating it into the model structure. This effectively improves the translation quality and enhances the translation performance of the model.

Different fusion methods are explored and analyzed to find the most suitable fusion method for memory information and global information.

Ablation experiments are designed to investigate the impact of key components on the model's performance. Comparative experiments are also designed to test the proposed model on a dataset in the field of electrical engineering and compare it with baseline models and other advanced models under the same experimental conditions, proving the superior performance of the proposed model.

2. Related Works

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 [8]. Mahsuli, MM proposed a method of modeling based on the length of the target sentence to improve Arabic-to-English translation [9]. 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 [10]. Laskar, SR improved English–Assamese machine translation through pretraining models and applied the pretrained multilingual context embedding alignment technology to the model, achieving good results [11]. Park, YH enhanced low-resource neural machine translation data through EvalNet and used data augmentation techniques to evaluate data quality [12]. 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 similarities between sentences and documents [13]. Gong, LC achieved good results on several low-resource datasets by guiding self-attention with syntactic graphs [14]. Hlaing, ZZ added an additional encoder to the transformer to introduce part-of-speech tagging, improving Thai-to-Myanmar, Myanmar-to-English, and Thai-to-English translation [15]. By assisting the transformer in learning from the corpus through prior knowledge, the translation model can learn more accurate and rich semantic knowledge during continuous training, thereby improving the accuracy of the translation results. However, the process of annotating and extracting prior knowledge from the corpus is complex and difficult, and integrating prior knowledge into the transformer often results in information incompatibility. Therefore, the focus of this paper is on structural improvements, attempting to integrate gated recur-

rent units to extract memory information from the data during training and avoid information disorder.

3. Methods

This paper improves upon the baseline model transformer proposed by Ashish Vaswani in 2017. The improved model consists of five parts: input layer, memory information encoder, global information encoder, decoder, and output layer. Compared with the traditional transformer, it can integrate the memory information from different time steps, capturing the complete past and future context information at the current time step in the input sequence, which makes up for the drawback of the transformer's inability to handle longer sequences. The global information encoder and decoder both consist of i = 6 identical layers stacked together, and the memory information encoder is also stacked with N identical layers. Absolute position encoding is used to obtain positional information for the source and target languages.

The improved model is shown in Figure 1. To ensure that memory information can be integrated into the transformer in a highly adaptive manner, we propose a memory information encoder based on the GRU [16] and integrate it into the right side of the global information encoder. In addition, we add a multi-head attention mechanism to the decoder unit to receive the output from the memory information encoder so that memory information and global information can be fused in the vector fusion layer.

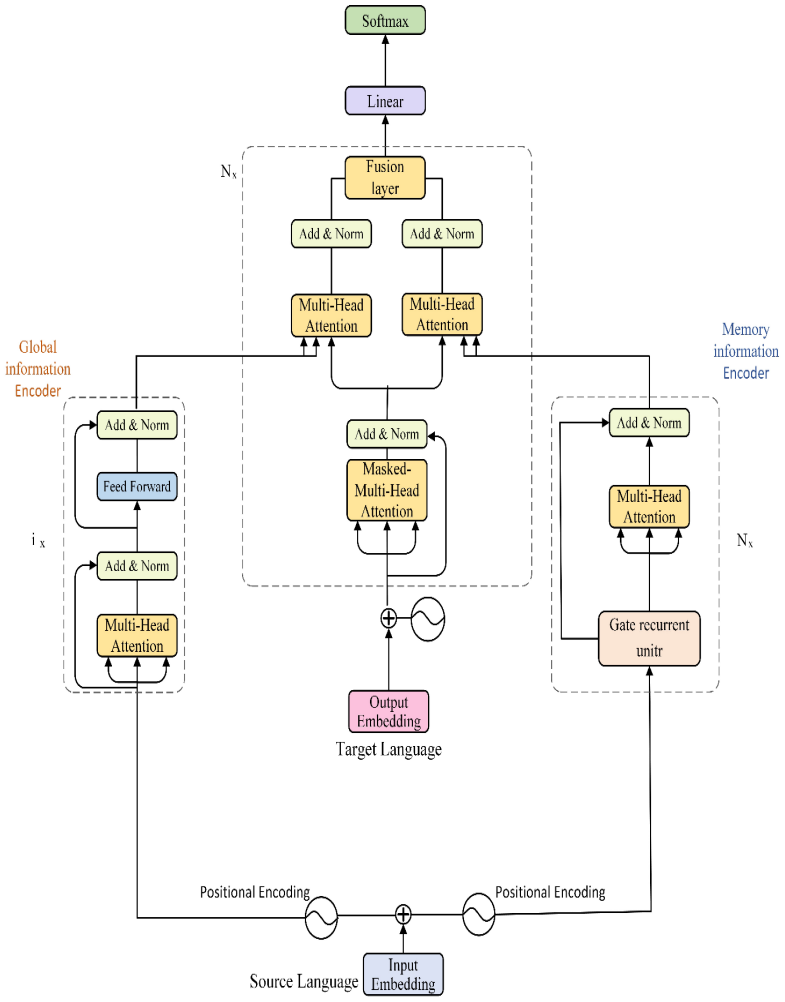

Figure 1. Transformer with fusion of memory information.

The main advantage of the transformer with fusion of memory information compared with other models is that it can directly extract memory information from the source language sequence through the memory information encoder without the risk of information incompatibility in the subsequent fusion process. If we first extract the memory information from the original corpus and then fuse it into the transformer, information disorder and incompatibility may occur during the fusion process, which not only wastes time but also has a negative impact on the model's performance. To maximize the model's performance, we propose four different fusion methods in Section 3.4 of the paper and conduct experiments and analysis on different fusion methods in Section 4.2, selecting the most suitable fusion method for integrating global and memory information.

3.1. Global Information Encoder

Each encoder consists of two sublayers: an attention layer and a feed-forward neural network. Both sublayers have residual connections and normalization for data regularization. The global information encoder takes S^{i-1} as the input vector, performs a global self-attention calculation on it (Equation (1)), integrates the extracted global information with the source language passed through residual connections, and then performs normalization (Equation (2)):

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

$$S_{out} = Addnorm(S_{self-attention} + S^{i-1}) \tag{2}$$

S^{i-1} represents the output of the i-th layer of the encoder. Each layer uses the output of the previous layer as the input of the next layer. The input of the first layer of the encoder is the source language after embedding and encoding. $Multihead()$ represents the multi-head attention mechanism, and $Addnorm()$ represents the residual connection and normalization.

3.2. Memory Information Encoder

The reason for building the memory information encoder on the basis of the GRU is that it not only can extract memory information but also has a simpler model structure and fewer model parameters. Compared with long short-term memory (LSTM) networks, the GRU consists of only an update gate and a reset gate, which can store memory information of longer sequences and not clear it over time. This mechanism can complement the transformer reciprocally, enabling the improved model to effectively learn memory information. The memory information encoder consists of a reset gate, an update gate, and an attention layer, with an internal structure as shown in Figure 2.

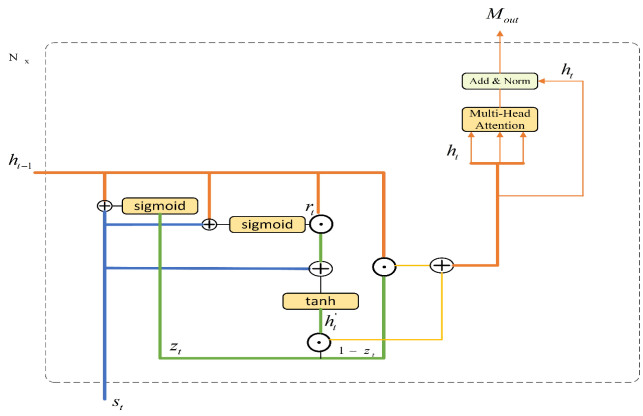

Figure 2. The internal structure of the memory information encoder.

3.2.1. Update Gate

The reset gate captures the memory information of the source language in a short period, determining how to combine new input information with previous memory information. If the reset gate is closed, historical information will be ignored to prevent irrelevant information from affecting future outputs. The update gate adds the linearly transformed s_t^{i-1} and h_{t-1}, and applies an activation function for a nonlinear mapping, compressing the result between 0 and 1 (Equation (3)),

$$Z_t = Sigmoid(W^{(z)}s_t + W^{(z)}h_{t-1}) \tag{3}$$

where s_t represents the input vector at time step t, and it undergoes a linear transformation (multiplied by the weight matrix $W^{(z)}$); h_{t-1} represents the information from the previous time step of $t-1$, which also undergoes a linear transformation; *Sigmoid* represents the activation function; and Z_t is the gating coefficient of the update gate, which controls the flow of memory information.

3.2.2. Reset Gate

The update gate defines the amount of previously stored memory at the current time, which extracts memory information over a longer period, controls the influence of historical information on the current time output, and passes the long-term memory information down. The calculation process of the reset gate is similar to that of the update gate, except that the parameters of the linear transformation (weight matrix) are different (Equation (4)),

$$R_t = Sigmoid(W^{(r)}s_t^{i-1} + W^{(r)}h_{t-1}) \tag{4}$$

where R_t is the gate coefficient of the reset gate, which determines the forgetting and retention of information. For example, if the gate value corresponding to an element is 0, it means that the information of this element will be completely forgotten. The s_t and h_{t-1} are multiplied, respectively, by the weight matrixes W and U. The R_t undergoes a Hadamard operation with the linearly transformed h_{t-1} to determine the current information that needs to be retained and the historical information that needs to be forgotten. The update gate adds the results of these two parts, and then applies a hyperbolic tangent activation function to nonlinearly map the addition result, obtaining the current memory information h'_t (Equation (5)),

$$h'_t = \tanh(Ws_t + R_t \odot Uh_{t-1}) \tag{5}$$

where $\odot$ represents the Hadamard product. Then, z_t and $(1-z_t)$ are separately Hadamard-multiplied with h_{t-1} and h'_t, respectively, to obtain the memory information retained from the previous time step t and the memory information retained from the current time step t, which are both kept until the final time step. Finally, these two parts of information are added to obtain the final integrated memory information h'_t (Equation (6)):

$$h_t = z_t \odot h_{t-1} + (1-z_t) \odot h'_t \tag{6}$$

3.2.3. Attention Layer

The attention layer consists of multiple attention mechanisms and normalization layers, which enable the model to fully learn the dependency relationship of memory information at different times through the attention calculation process, ensuring that the model can obtain compatible memory information.

After obtaining the memory information, it undergoes an attention calculation through a multi-head self-attention mechanism. The resulting attention weights are then multiplied by the memory information and summed up to obtain the attention output. The attention output is then added to the output of the residual network to obtain the result of the memory information encoder (Equation (7)),

$$M_{memory-attention} = Multihead(h_t, h_t, h_t) \tag{7}$$

$$M_{out} = Addnorm(M_{memory-attention} + h_t) \tag{8}$$

where $M_{memory-attention}$ represents the output of the multi-head self-attention mechanism, and M_{out} represents the output of the memory information encoder.

3.3. Decoder

Because the memory information needs to be fed into the decoder, a multi-head attention mechanism needs to be added to the original decoder structure to receive the output from the memory information encoder. In addition, a vector fusion layer needs to be added after the attention layer for the fusion of global and memory information. Therefore, the improved decoder consists of four parts: a masked self-attention sublayer, an attention sublayer composed of a context-decoder and memory-decoder, a vector fusion layer, and a fully connected feed-forward network sublayer. The internal structure of masked-decoder, encoder-decoder, and memory-decoder is the same, and they all use dot product for calculation. The main difference lies in the query vector Q, key vector K, and value vector V.

The masked-decoder is responsible for learning the dependency relationship between the target language sequences and preparing for the association between the target language and memory information/global information. The mask mechanism inside is to prevent the overfitting caused by the model using the results of the previous decoder round during training. Because the masked-decoder performs attention calculation only on the target language, its Q, K, and V vectors are T^{i-1}, and the calculation process is as follows:

$$T_{Masked-Decoder} = Addnorm(MaskedMultihead(T^{i-1}, T^{i-1}, T^{i-1})) \tag{9}$$

T^{i-1} represents the output of the i-th layer of the decoder, and for each layer, the output is the previous layer of the decoder. $MaskedMultihead()$ represents the masked attention mechanism.

In the encoder-decoder, the output of the masked self-attention sublayer is used as the Q vector, the output of the global information encoder is used as the K and V vectors, and through the attention calculation process, the model learns the dependency relationship between the target language sequence and the source language sequence during training, further establishing the contextual information correlation:

$$T_{Encoder-Decoder} = Addnorm(Multihead(T_{self-decoder}, S_{out}, S_{out})) \tag{10}$$

The main function of the memory-decoder is to receive the output from the memory information encoder, establish the association between the memory information and the target information through the attention mechanism, and ensure that the model can learn semantic knowledge containing memory information during training. Therefore, the memory-decoder uses the output of the masked attention sublayer as the Q vector and the output of the memory information encoder as the K and V vectors, and the specific calculation process is as follows:

$$T_{Memory-Decoder} = Addnorm(Multihead(T_{self-decoder}, M_{out}, M_{out})) \tag{11}$$

3.4. Fusion Method

After obtaining the decoded global information $T_{Encoder-Decoder}$ and the memory information $T_{Memory-Decoder}$, the two types of decoded information need to be fused in the fusion layer. This article compares and selects four fusion methods, including attention fusion, balance coefficient fusion, concatenation fusion, and arithmetic average fusion, and selects the most effective one as the final fusion method for the model. The following sections analyze these four fusion methods, and the final experimental results are shown in Table 1.

Table 1. BLEU scores for different fusion methods.

Model	BLEU/%	σ
Baseline	34.25	–
Our + mean	35.48	1.23↑
Our + gate	35.82	1.57↑
Our + attention	35.18	0.93↑
Our + cat	36.29	2.04↑

where: σ = (BLEU value of our model—BLEU value of baseline model) × 100; "Our" represents our model; and "mean", "gate", "attention", and "cat" represent arithmetic average fusion, gate fusion, attention fusion, and concatenation fusion, respectively.

3.4.1. Attention Fusion

The attention mechanism in deep learning is a method that imitates the human visual and cognitive systems, allowing neural networks to focus on relevant parts of the input data. The basic idea of the attention mechanism is that each element in the sequence data can establish a relationship with other elements in the sequence, not just rely on adjacent elements:

$$F_{attention} = Addnorm(MultiHead(T_{Encoder-Decoder}, T_{Memory-Decoder}, T_{Memory-Decoder})) \quad (12)$$

Using the attention mechanism to fuse global information and memory information can allow the model to assign different weights to different positions of the input sequence, adaptively capture the dependency relationship between global information and memory information by calculating the relative importance between elements, and obtain the most important semantic information. Setting the Q vector in the multi-head attention mechanism as $T_{Encoder-Decoder}$, and the K and V vectors as $T_{Memory-Decoder}$, the attention calculation is performed to obtain the fused vector.

3.4.2. Gate Fusion

Influenced by previous works [17–19], first, $T_{Encoder-Decoder}$ and $T_{Memory-Decoder}$ are concatenated along the last dimension of $T_{Encoder-Decoder}$. Then, the $Sigmoid$ function is used to normalize them and obtain the gate coefficient y. $W_{\partial t}$ and $b_{\partial t}$ are parameters, and y is a number between 0 and 1.

$$y = Sigmoid(W_{\partial t}([T_{Encoder-Decoder} : T_{Memory-Decoder}] + b_{\partial t}), \dim = -1) \quad (13)$$

Finally, a simple weighted operation is performed on $T_{Encoder-Decoder}$ and $T_{Memory-Decoder}$ to obtain the fused vector of the fusion layer,

$$F_{gate} = y * T_{Encoder-Decoder} + (1 - y) * T_{Memory-Decoder} \quad (14)$$

3.4.3. Concatenation Fusion

Concatenation fusion is a relatively simple process. M_{out} is concatenated along the last dimension of S_{out}, and then a linear layer is used to process the concatenated vector's dimension to prevent it from affecting the subsequent calculation process:

$$F_{cat} = W(cat([T_{Encoder-Decoder} : T_{Memory-Decoder}], \dim = -1)) \quad (15)$$

where $W()$ represents the linear layer used to process and update the parameters' dimensions, and $cat()$ represents the concatenation fusion, with "b" concatenated along the last dimension.

3.4.4. Arithmetic Average Fusion

Arithmetic average fusion is a method that takes the arithmetic mean of $T_{Encoder-Decoder}$ and $T_{Memory-Decoder}$ to obtain the fused vector. This method can make the calculation re-

sults smoother and reduce the overfitting phenomenon. Considering that global information and memory information should be fused with an appropriate and uniform proportion, in this section, the means of the output from the six layers of the global information encoder and the means of the output from the two layers of the memory information encoder are fused, and the calculation results are as follows:

$$S_{mean} = \frac{T_{Encoder-Decoder}}{6} \tag{16}$$

$$M_{mean} = \frac{T_{Memory-Decoder}}{6} \tag{17}$$

$$F_{mean} = Cat(S_{mean} : M_{mean}, \dim = -1) \tag{18}$$

where S_{mean} represents the mean of $T_{Encoder-Decoder}$, $T_{Memory-Decoder}$ represents the mean of the memory information, and F_{mean} represents the result of the arithmetic average fusion of S_{out} and M_{out}.

4. Experiment

In this part, we conducted experiments and research about the models proposed in this paper on the Chinese–English parallel corpus in the field of electrical engineering.

4.1. Dataset and Parameter Settings

As this article focuses on low-resource neural machine translation in the field of electrical engineering, the data used in the experiment must have strong specificity and professionalism. Therefore, we collected 190,000 parallel bilingual corpora from professional books [20–23], equipment manuals, references, and related papers in the relevant field as the dataset for the experiment.

We used the open-source system OpenNMT [24] to implement the baseline model transformer. Regarding data preprocessing, we limited the sentence length in the corpus to within 100, meaning that sentences longer than 100 were filtered out. The vocabulary size of both the source and target languages was set to 44,000. Jieba and NLTK were used for Chinese and English word segmentation, respectively. The training and testing sets each contained 2000 pairs of bilingual parallel sentences. During the training process, the word vector dimension and the hidden layer dimension of the encoder and decoder were set to 512, the batch size was set to 64, the Adam optimization algorithm was used, and the dropout probability of the neurons was set to 0.1. A total of 25,000 steps was trained in this experiment, and the model was validated every 1000 steps. The beam search method was used during decoding, with a beam size of five and the remaining parameters using the default settings of OpenNMT. All parameters in the experiment were consistent, and the translation results were evaluated using BLEU [25]. The input and output channels of the GRU were both set to a dimension of 512. This means that the input and output vectors are both 512-dimensional, ensuring consistency and compatibility. The choice of this dimension was based on experimental considerations and the specific requirements of our model architecture. The number of layers in the GRU was kept consistent with the number of layers in the encoder. This ensured that the information flow and transformations between the layers remained synchronized throughout the model. The activation functions used for the update gate and reset gate in the GRU were sigmoid and tanh, respectively. The sigmoid function was employed to calculate the update gate, which determines how much of the previous hidden state should be retained, while the tanh function was utilized for the reset gate, which controls how much of the previous hidden state should be forgotten.

4.2. Impact of Fusion Methods

To select the most suitable fusion method for the GRU, we experimented with models using concatenation fusion, gate fusion, attention fusion, and arithmetic average fu-

sion, and compared them with the baseline model. The experimental results are shown in Table 1, and the column chart of training data of different fusion methods is shown in Figure 3. (The horizontal axis represents the training rounds, while the vertical axis represents the BLEU value.)

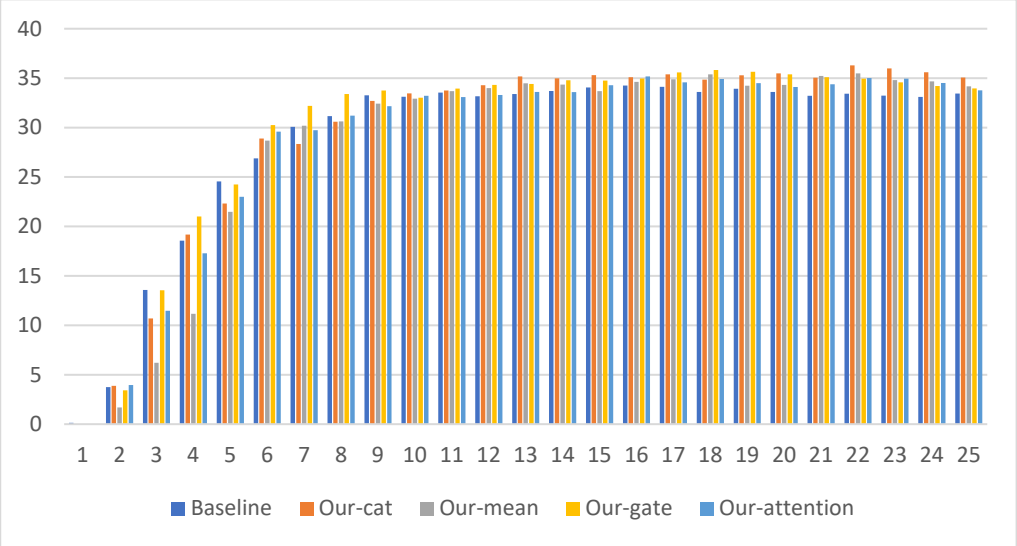

Figure 3. The bar chart of training data.

The baseline model achieved a BLEU score of 34.25, whereas all the models incorporating memory information achieved higher scores. The concatenation fusion method had the highest BLEU score of 36.29, which is 2.04 points higher than the baseline. The arithmetic average fusion, gate fusion, and attention fusion methods also outperformed the baseline model, but to a lesser extent.

These results suggest that incorporating memory information into the transformer model can improve translation performance, and the most effective way to do so is through concatenation fusion. Through our exploration and analysis, we believe that the reason for this result may be that the other three fusion methods may lose some useful information when fusing global information and memory information. Arithmetic average fusion loses some information when calculating the means of the two types of information. Gate fusion loses information when controlling and selecting the incoming information through the gate coefficient. Attention fusion discards some information based on the weight of the memory information vector. Concatenation fusion concatenates the memory information vector directly to the last dimension of the global information vector, reducing the possibility of losing information. Therefore, the model using the concatenation fusion method has better performance than the other three models.

4.3. Ablation Experiment

To demonstrate the effectiveness of the memory information, we conducted an ablation experiment on models with memory information encoders of different depths, and the experimental results are shown in Table 2.

Table 2. BLEU scores for models with memory information encoders of different depths.

Model	BLEU/%	σ
Baseline	34.25	–
Our methods-layer = 1	34.66	0.41↑
Our methods-layer = 2	34.82	0.57↑
Our methods-layer = 3	35.07	0.82↑
Our methods-layer = 4	35.48	1.23↑
Our methods-layer = 5	35.93	1.68↑
Our methods-layer = 6	36.29	2.04↑

The results are shown in Figure 4 and demonstrate that the BLEU score increases with the number of layers in the memory information encoder, reaching a maximum of 36.29 when the maximum number of layers is used. This suggests that incorporating memory information into the transformer model can improve translation performance. These results also highlight the importance of memory information in the transformer model and support the effectiveness of the proposed method for incorporating memory information through integrating the memory information encoder.

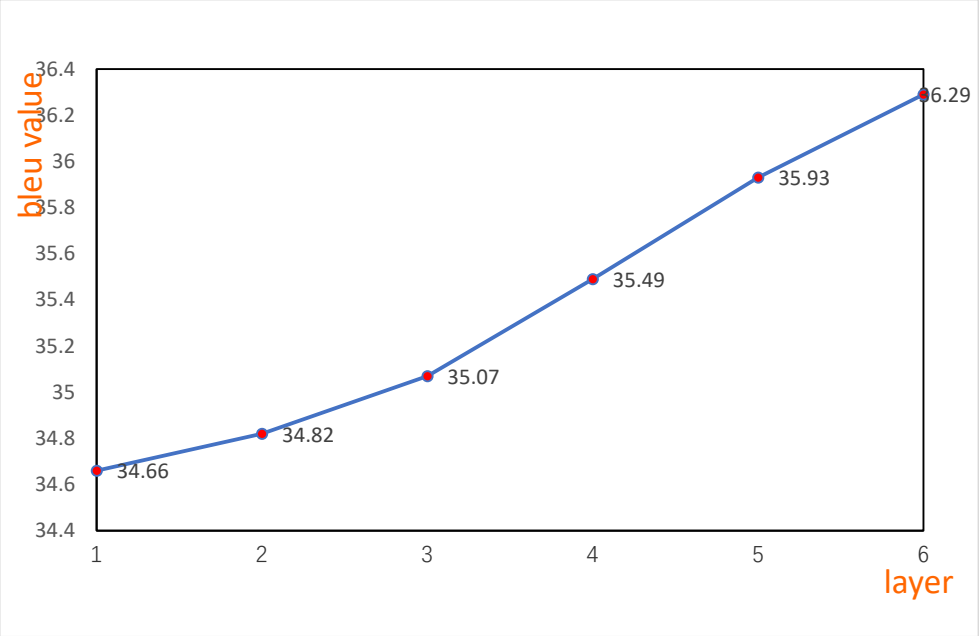

Figure 4. The curve plot of the BLEU value.

4.4. Comparative Experiment

To further demonstrate the effectiveness of our method, we conducted comparative experiments with baseline models, vector fusion [26], and key information fusion [27] on a dataset in the field of electrical engineering. The experimental conditions were kept consistent, and the results are shown in Table 3 and the translation samples are shown in Table 4.

Table 3. BLEU scores for comparative experiment.

Model	BLEU/%	σ
Baseline	34.25	–
Vector fusion	35.83	1.49↑
Key information fusion	34.97	0.72↑
Our methods	36.29	2.04↑

Vector fusion: proposed in 2023 by Hong Chen et al., this method improves low-resource neural machine translation by using weight fusion.

Key information fusion: proposed in 2023 by Shije Hu et al., this method uses a dual encoder structure to integrate key information from the text into the transformer, thereby improving its performance.

Based on the experimental results provided, it appears that the proposed method outperforms the baseline and the other two comparative methods, vector fusion and key information fusion. The baseline method achieved an accuracy of 34.25, while the vector fusion achieved an accuracy of 35.83, an improvement of 1.49. The key information fusion achieved an accuracy of 34.97, an improvement of 0.72. In contrast, the proposed method achieved the highest accuracy of 36.29, an improvement of 2.04 compared with

the baseline. These results suggest that the proposed method is more effective than the other methods in improving the performance of the model on the given dataset in the field of electrical engineering.

Table 4. Translation samples.

Source text	A design based on a microcontroller for a digital pulse counting module and digital display module was developed, and a design proposal for the hardware circuit and software program of the instrument was presented.
Reference	设计了基于单片机的数字脉冲 计数 模块和数字显示模块，并提出了该仪表的硬件电路和软件程序的设计方案。
Translation of **our model**	
设计了基于单片机的数字脉冲计数和数字显示模块，提出了该电表的硬件电路和软件程序的设计方案。	
Translation of **Vector fusion**	
在单片机的基础上设计了数字脉冲计数和数字显示模块，给出了该电表的硬件和软件的设计。	
Translation of **Key information fusion**	
设计了关于单片机的数字脉冲计数器和数字显示板块，提出了该电表的硬件和软件程序的电路设计。	

4.5. Experimental Analysis

Based on the analysis of the results from the ablation experiments, fusion method experiments, and comparative experiments, the following conclusions can be drawn:

Concatenation fusion is the most suitable fusion method among the four fusion approaches for the proposed method in this paper. It minimizes the loss of information during the fusion of global and memory information, enabling the transformer to obtain more rich and beneficial semantic information.

Memory information is beneficial for the translation task of the transformer. It helps the transformer learn sequence information from different time steps, strengthens the model's understanding of dependencies between sequences, and improves the overall performance of the model.

Compared with the vector fusion model and the key information fusion model, the proposed model in this paper demonstrates superior performance on electrical engineering datasets. It further assists the transformer in translating specialized electrical language into the corresponding target language. The results of both the ablation experiments and the comparative experiments prove that the transformer integrated with the memory information encoder can achieve further improvements in translation accuracy with the help of memory information.

Overall, the integration of the memory information encoder into the transformer enhances its performance, particularly in the domain of electrical engineering datasets. It enables the transformer to capture and utilize memory information effectively, leading to improved translation accuracy.

5. Conclusions

To address the poor performance of the transformer model on electrical engineering datasets, this paper proposes a memory information encoder based on the GRU and integrates it into the overall structure of the transformer. The GRU is capable of extracting memory information from the source language sequence, and the attention mechanism learns from the extracted information. By combining the advantages of the GRU and the attention mechanism, the improved model enhances its performance and compensates for the baseline model's weakness in translating complex and lengthy sentences.

To ensure that the transformer can effectively fuse global and memory information in the fusion layer of the decoder, this paper explores and analyzes four fusion methods. Ultimately, the concatenation fusion method is selected as the fusion approach for global

and memory information, laying a solid foundation for the model to acquire richer semantic knowledge. The results of the ablation experiments demonstrate that the model integrated with the memory information encoder improves the translation quality of the transformer and steadily enhances the model's performance. The results of comparative experiments further validate the effectiveness of the proposed method, showing that the model proposed in this paper is not inferior to the comparative models.

The experimental results confirm the effectiveness of the proposed method and model, achieving a maximum improvement of 2.04 percentage points in the BLEU score compared with the baseline model on electrical engineering datasets. Compared with other methods, integrating memory information into the transformer ensures that the extracted memory information is compatible and complementary to the global information, saving time and effort required for data processing.

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

Funding: This research was funded by J.Z., grant number U2004163, and the APC was funded by J.Z.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets are not published. Please contact the author if necessary.

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

References

1. Karyukin, V.; Rakhimova, D.; Karibayeva, A.; Turganbayeva, A.; Turarbek, A. The neural machine translation models for the low-resource Kazakh–English language pair. *PeerJ Comput. Sci.* **2023**, *9*, e1224. [CrossRef] [PubMed]
2. Maučec, M.S.; Donaj, G. Machine translation and the evaluation of its quality. *Recent Trends Comput. Intell.* **2019**, 143.
3. Kalchbrenner, N.; Blunsom, P. Recurrent convolutional neural networks for discourse compositionality. *arXiv* **2013**, arXiv:1306.3584.
4. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015.
5. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the ICML, Sydney, Australia, 6–11 August 2017.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762, 2017.
7. Araabi, A.; Monz, C. Optimizing transformer for low-resource neural machine translation. *arXiv* **2020**, arXiv:2011.02266.
8. 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]
9. Mahsuli, M.M.; Khadivi, S.; Homayounpour, M.M. LenM: Improving Low-Resource Neural Machine Translation Using Target Length Modeling. *Neural. Proc. Lett.* **2023**, 1–32. [CrossRef]
10. Pham, N.L.; Pham, T.V. A Data Augmentation Method for English-Vietnamese Neural Machine Translation. *IEEE Access* **2023**, *11*, 28034–28044. [CrossRef]
11. 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]
12. 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]
13. 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, 13 June 2022; pp. 4933–4943.
14. 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. [CrossRef]
15. 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]
16. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.

17. Gulcehre, C.; Firat, O.; Xu, K.; Cho, K.; Barrault, L.; Lin, H.C.; Bengio, Y. On using monolingual corpora in neural machine translation. *arXiv* **2015**, arXiv:1503.03535.
18. Wang, Y.; Xia, Y.; Tian, F.; Gao, F.; Qin, T.; Zhai, C.X.; Liu, T.Y. Neural machine translation with soft prototype. *Adv. Neural Informat. Process. Syst.* **2019**, *32*.
19. 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–4 November 2018; pp. 3042–3047.
20. Bimal, K. *Modern Power Electronics and AC Drives*; Prentice-Hall: Hoboken, NJ, USA, 2001.
21. Bimal, K. *Modern Power Electronics and AC Drive*; Wang, C.; Zhao, J.; Yu, Q.; Cheng, H., Translators; Machinery Industry Press: Beijing, China, 2005.
22. Wang, Q.; Glover, J.D. *Power System Analysis and Design (Adapted in English)*; Machinery Industry Press: Beijing, China, 2009.
23. 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.
24. Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; Rush, A.M. Opennmt: Open-Source Toolkit for Neural Machine Translation. *arXiv* **2017**, arXiv:1701.02810.
25. 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 (2002), Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
26. Chen, H.; Chen, Y.; Zhang, J. Neural Machine Translation of Electrical Engineering Based on Vector Fusion. *Appl. Sci.* **2023**, *13*, 2325. [CrossRef]
27. Hu, S.; Li, X.; Bai, J.; Lei, H.; Qian, W.; Hu, S.; Yang, S. Neural Machine Translation by Fusing Key Information of Text. *CMC-Comput. Mater. Cont.* **2023**, *74*, 2803–2815. [CrossRef]