applied sciences

MDPI

Article

Neural Machine Translation Research on Syntactic Information Fusion Based on the Field of Electrical Engineering

Yanna Sang [1,2], Yuan Chen [3] and Juwei Zhang [1,2,*]

[1] School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China
[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
* Correspondence: juweizhang@haust.edu.cn

Abstract: Neural machine translation has achieved good translation results, but needs further improvement in low-resource and domain-specific translation. To this end, the paper proposed to incorporate source language syntactic information into neural machine translation models. Two novel approaches, namely Contrastive Language–Image Pre-training(CLIP) and Cross-attention Fusion (CAF), were compared to a base transformer model on EN–ZH and ZH–EN pair machine translation focusing on the electrical engineering domain. In addition, an ablation study on the effect of both proposed methods was presented. Among them, the CLIP pre-training method improved significantly compared with the baseline system, and the BLEU values in the EN–ZH and ZH–EN tasks increased by 3.37 and 3.18 percentage points, respectively.

Keywords: neural machine translation; graph neural network; syntax parsing tree; transformer

Citation: Sang, Y.; Chen, Y.; Zhang, J. Neural Machine Translation Research on Syntactic Information Fusion Based on the Field of Electrical Engineering. *Appl. Sci.* **2023**, *13*, 12905. https://doi.org/10.3390/app132312905

Academic Editors: Christos Bouras and Andreas Sumper

Received: 10 October 2023
Revised: 6 November 2023
Accepted: 30 November 2023
Published: 1 December 2023

1. Introduction

Machine translation [1] is the process of automatic translation using computers, and it is an important research direction in the fields of natural language processing and artificial intelligence. Due to extensive differences in grammar, vocabulary, and cultural background among different languages, and the increasing demand for accuracy and professionalism in translation, traditional rule-based and statistical methods [2] are no longer sufficient. Neural machine translation (NMT) [3] offers a more effective solution. Neural machine translation, trained on large-scale corpora and deep neural network models, can better meet the requirements of accuracy and expertise in translation within specialized domains.

Each sentence in a language is not a simple list of words, but rather is constrained by the grammatical structure of the language. Neural machine translation models often overlook the syntactic structure information of sentences during translation, leading to issues such as mistranslation, over-translation, and omission in the translated results. Therefore, we believe that incorporating syntactic structure information into neural machine translation models can effectively handle ambiguity and complex language structures, resulting in more accurate translation outputs. The syntactic tree structure is illustrated in Figure 1.

There are many ways to integrate syntactic information into neural machine translation: for example, Chen et al. [4] proposed using a bidirectional tree encoder to learn structural representations of trees. The tree coverage model enables the model's attention to depend on the source-side syntactic information, thereby enhancing the translation accuracy. However, this approach can lead to increased complexity in the neural network structure, resulting in lower training efficiency. Eriguchi et al. [5] proposed a model for performing neural machine translation with conversion from strings into trees, which translates the source sentence into a linearized, lemmatized constituency tree and enriches the output with syntactic information. The disadvantage of this approach is that the Tree-LSTM

network structure is relatively complex. Xin Zheng and Hailong Chen [6] introduced dependency parsing and Long Short-Term Memory (LSTM) networks to construct the syntactic structure information of the source language in a neural machine translation system. Sufeng Duan et al. [7] proposed a method to linearize syntactic information and incorporate it into the input embedding layer, as well as encode the depth of the tree into the position encoders of the transformer. This approach only requires simple modifications to the model's architecture. However, the linearized sequences are often long, resulting in decreased training efficiency. Tamura and Ma [8] proposed a neural machine translation method based on neural syntactic distance, which requires a large amount of training data to effectively train the neural syntactic distance model, and also incurs a high computation cost in calculating syntactic distance. Bugliarello et al. [9] proposed a novel parameter-free local self-attention mechanism, which has shown good performance in handling long sentences and low-resource scenarios. But, the methods of Gong [10] and Bugliarello et al. require a large amount of training data and are also limited by the parser. Most of these methods utilize Tree-LSTM [11], directly encode syntactic information in the tree structure, or represent the syntactic tree using linearization [12]. While they have an improved translation performance, they often require a large amount of training data and incur additional computational costs.

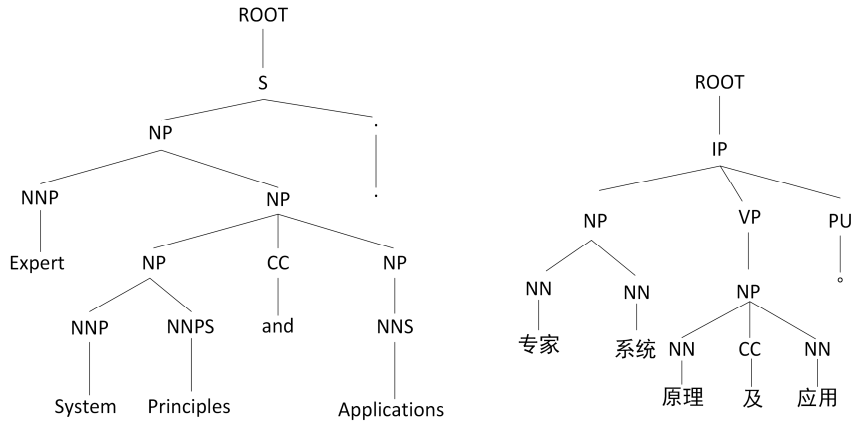

Figure 1. EN and ZH syntax parsing tree structure diagram. 专家系统原理及应用 (Expert System Principles and Applications).

We propose two novel methods of integrating syntactic information into NMT. The first method involves pre-training using the CLIP [13] model, where the text encoder and graph neural network are used to extract information from both the text and syntactic tree modalities, and the obtained information is then passed into the transformer model for machine translation. The second method involves cross-attention fusion, where we use graph neural networks (GNNs) [14] and transformers separately to process syntactic tree graphs and text data, respectively, and then fuse the two data sources using a cross-attention mechanism to complete the machine translation task. Our proposed methods effectively overcome the drawbacks of the existing approaches.

We use an electrical domain corpus for model training, and use the Stanford Parser for syntax analysis to obtain a syntax tree and convert it into a graph structure. Inspired by multimodal neural machine translation [15], we use GNNs to extract the grammatical information of sentences from the syntactic tree graph, and integrate the grammatical information of sentences into neural machine translation. The main contributions of this paper are as follows:

1. This paper proposes two approaches to incorporate syntactic structure information into neural machine translation.

2. This paper compares the proposed method with the standard transformer, and analyzes the impact of sentence complexity on the translation results of the translation model.
3. We adopt a research method of neural machine translation based on an electrical corpus, which fully utilizes the professional terminology and language conventions in the field of electrical engineering, in order to meet the practical application requirements. This approach results in machine translation that better aligns with the professional requirements.

2. Related Methods

2.1. Graph Neural Networks

Graph neural networks (GNNs) [14] are a type of deep learning model used for processing graph-structured data. The core idea of GNNs is to learn representations for each node by propagating and aggregating information from the nodes and edges of the graph. GNNs can automatically learn complex relationships between nodes and the global structure of a graph without relying on manually engineered feature engineering. Due to their strong performance, interpretability, and powerful expressive capabilities for graph structures, GNNs have recently emerged as a practical method for graph analysis. They are widely applied in various domains such as social network analysis, recommendation systems, image and vision tasks, molecular chemistry, text and natural language processing, and more.

A graph is composed of many nodes and edges, and each node is defined by its own characteristics and the characteristics of the nodes connected to it. When we visualize the syntax tree, it becomes a graph, as shown in Figure 1. The structural information of the syntax tree can be extracted by traversing the input syntax tree graph structure using a GNN. Nodes are established to represent each word or piece of punctuation in the input, and edges represent the relationships between them. GNNs can extract the dependency relationship between different nodes in the syntax tree and effectively encode the structure information of the syntax tree into a vector, facilitating better integration into the neural machine translation model. A graph can be represented by a set V of vertices or nodes and a set E of edges, namely:

$$G = (V, E) \tag{1}$$

where v_i belongs to V to represent a node, and e_{ij} represents an edge from v_j to v_i, as shown in Figure 2. $e_{ij} = (v_i, v_j) \in E$. A is an adjacency matrix and $A \in R^{n \times n}$, when $e_{ij} \in E$, then $A_{ij} = 1$; when $e_{ij} \notin E$, then $A_{ij} = 0$.

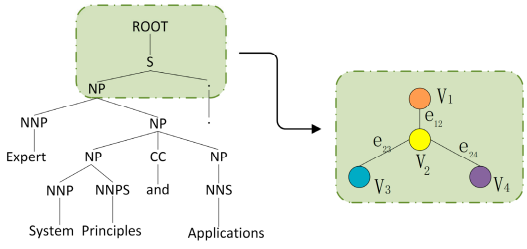

Figure 2. Schematic diagram of edges and nodes of syntactic tree graph.

A GNN iteratively updates the feature vectors of nodes. In each iteration, the feature vector of a node interacts and aggregates information with the feature vectors of its neighboring nodes to capture the dependency relationships between nodes. The model formula expression for a GNN is as follows:

In each iteration, for node v, its updated feature vector can be expressed as:

$$h_v^{(t+1)} = \text{AGGREGATE}\left\{ f(h_v^{(t)} + N_v) \right\} \tag{2}$$

$$N_v = \sum_{j \in N_i} W_j \bullet x_j \tag{3}$$

Among them, $h_v^{(t)}$ represents the feature vector of node v after the t-th round of iterations, and N_v represents the feature vector set of neighbor nodes of node v. x_j represents the feature vector of the j-th neighbor node, and W_j is the weight matrix. AGGREGATE represents an aggregation operation, the purpose being to merge the information of neighbor nodes into the feature vector of node v to reflect the relationship between node v and its neighbor nodes. f represents an update function used to update the feature vector of node v, with which can interact and aggregate information between the feature vector of node v and the feature vectors of its neighbor nodes. The illustration of GNN aggregation operation is shown in Figure 3.

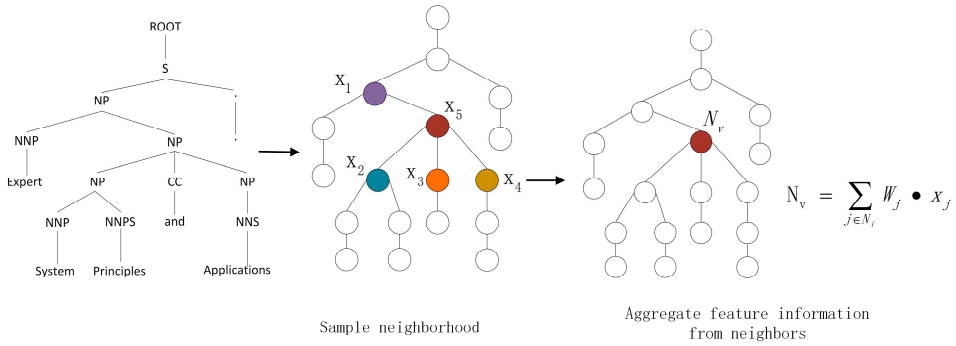

Figure 3. Schematic diagram of GNN aggregation operation.

Using multiple iterations, the GNN model can progressively update the feature vectors of nodes and extract information between different nodes in the syntactic tree.

2.2. Transformer

A transformer is a neural network model for sequence-to-sequence learning. Since its introduction by Vaswani [16] and others in 2017, the transformer has achieved great success in the field of natural language processing. It has since expanded to other fields such as image processing [17–19], speech processing [20,21], and more. Currently, the most popular ChatGPT [22,23] language model is trained based on the transformer architecture. The transformer model is unique in that it only uses self-attention mechanisms and standard feed-forward neural networks, without relying on any recurrent units or convolution operations. The advantage of the self-attention mechanism is that it can directly model the relationship between any two units in the sequence, thus effectively solving the problem of long-distance dependence.

Figure 4 shows the structure of a transformer. Both the encoder and the decoder are composed of several layers with the same structure, but the parameters are not shared. Each layer of the encoder consists of two sub-layers, followed by a self-attention sub-layer and a feed-forward network sub-layer. Each sub-layer has a residual link followed by a layer normalization operation. The decoder also has six layers, but with three sublayers and an attention sublayer (which computes attention on the encoder output). Figures 5 and 6 show the difference between the inputs of two different attention sublayers, respectively. The input of each layer of the encoder and decoder is a sequence of vectors, and the output is a sequence of vectors of the same size.

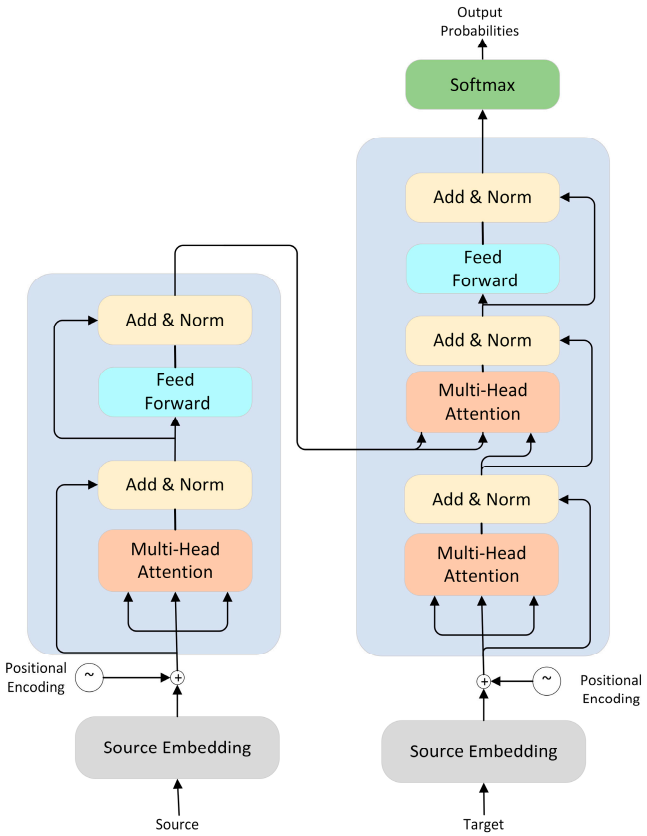

Figure 4. Transformer model structure.

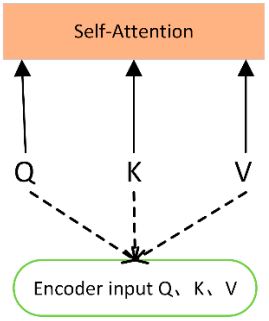

Figure 5. Input of self-attention model.

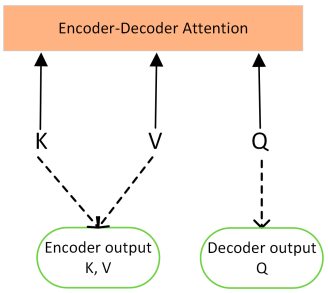

Figure 6. The input of the encoder–decoder attention model.

Given a sentence $x = \{x_1, x_2, x_3, \ldots, x_L\}$ at the source language end, where L represents the length of the input sentence at the source end, there are three weight matrices to generate the query (Q), key (K), and value (V) after the word embedding layer and position encoding layer, and the value (V) after the word embedding layer and position

encoding layer. The dimension of Q and K is $L \times d_k$, and the dimension of V is $L \times d_v$; d_k and d_v represent the size of the query and value, respectively. After obtaining Q, K, and V, the attention operation can be performed. The attention operation formula is:

$$\text{Attention}(H) = \text{softmax}(\frac{QK^{\text{T}}}{\sqrt{dk}}) \tag{4}$$

The sub-layers between the encoder and the decoder use residual connections, and the formula for residual connections is:

$$h^l = h^{l-1} + f_{sl}(h^{l-1}) \tag{5}$$

Among them, h^l represents the output of the first sublayer, and $f_{sl}(h^{(l-1)})$ represents the functional function of this layer.

3. Model

3.1. CLIP Pre-Training Fusion

CLIP (Contrastive Language-Image Pretraining) is a pretraining model developed by OpenAI, aiming to build a shared semantic feature space by learning the contrastive relationship between images and text. The design of CLIP is based on the idea that images and text should be treated as equally important inputs, and their semantic relationship can be established via joint training. Traditional computer vision models typically convert images into vector representations for tasks like image classification and detection. However, this approach fails to fully utilize the rich information within images and the interplay between images and text. CLIP innovatively treats images and text on equal footing, training the model to understand their semantic relationship using joint learning.

Figure 7 shows the model architecture of CLIP used in this paper for pre-training. The CLIP model consists of two encoders, the text encoder and the graph encoder, which are responsible for processing text and image data, respectively. The text encoder utilizes a transformer model to transform the input text data into a high-dimensional text feature vector (T_N) representation. Our graph encoder employs a GNN model to learn and represent the syntactic information within the input graphs and convert it into a high-dimensional graph feature vector (I_N) representation.

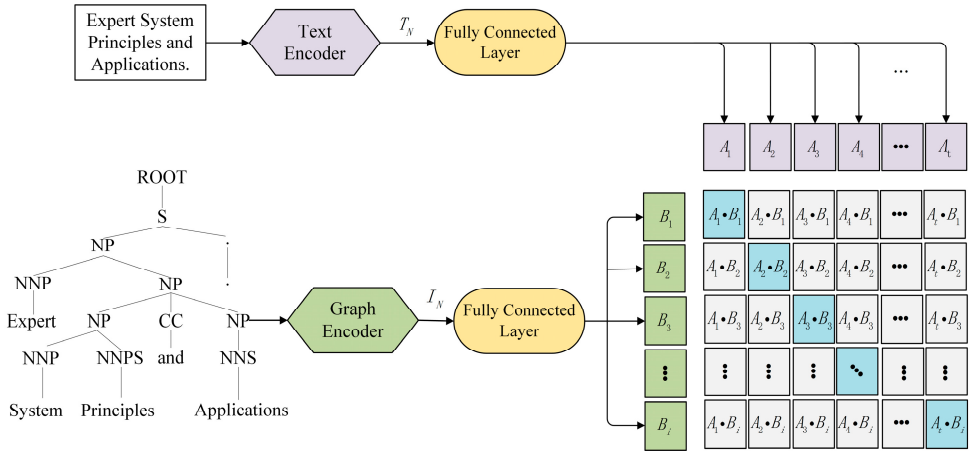

Figure 7. CLIP model architecture.

Next, we use a fully connected layer as the mapping function to map the text feature vector (T_N) and the graph feature vector (I_N) to a shared multimodal feature space. We aim to map them to a feature space of dimension D.

First, we apply a linear transformation to the text feature vector (T_N) to map it to a feature space of dimension D. This can be achieved using a weight matrix (W_t) and a bias vector (b_t), represented as:

$$Z_t = W_t \times T_N + b_t \tag{6}$$

Among them, Z_t is the intermediate feature vector after linear transformation of the text feature vector.

Next, we apply the ReLU non-linear activation function to the intermediate feature vector (Z_t) to introduce non-linear transformation. This can be expressed as:

$$A_t = \mathrm{ReLU}(Z_t) \tag{7}$$

Among them, A_t is the text feature vector after passing the non-linear activation function.

Similarly, we perform similar linear transformation and non-linear activation function operations on the graph feature vector (I_N) to obtain the intermediate feature vector of the graph:

$$Z_i = W_i \times I_N + b_i \tag{8}$$

$$A_i = \mathrm{ReLU}(Z_i) \tag{9}$$

Among them, Z_i is the intermediate feature vector after linear transformation of the graphic feature vector, and A_i is the graphic feature vector after going through the nonlinear activation function.

Finally, we combine the non-linearly transformed text feature vector (A_t) and graph feature vector (A_i) to obtain the shared multimodal feature vector. This is achieved by a linear transformation using an additional weight matrix (W_m) and bias vector (b_m), represented as:

$$M = W_m \times [A_t : B_i] + b_m \tag{10}$$

Among them, $[A_t : B_i]$ means splicing the text feature vector and the graphic feature vector in the feature dimension to obtain a vector with a 2D dimension. M represents the final multimodal feature vector.

In this way, the GNN model and the transformer model can mutually incorporate information from both modalities, thereby better capturing the correlations and semantics between text and graphics. Finally, we fine-tune the transformer model, which incorporates information from both modalities, for machine translation tasks. This approach leverages the correlation and semantic information between the text and graphics to improve the quality and accuracy of machine translation, bringing new ideas and methods to the development of machine translation technology.

3.2. Cross-Attention Fusion

The conventional neural machine translation model first encodes the input x of the source language into a hidden layer vector, then the system obtains the context information z_t of the source language and the target using the attention mechanism, and finally, the decoder predicts the output y based on the hidden layer information of the target language, as shown in Figure 8.

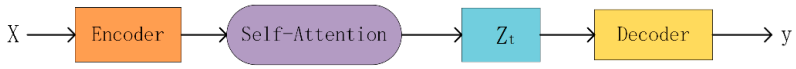

Figure 8. Conventional neural machine translation model.

In this section, we propose a neural machine translation approach called cross-attention fusion (CAF), as shown in Figure 9. Cross-attention fusion is a method that integrates syntactic structure information into the neural machine translation model to enhance its modeling capability for both semantic and syntactic information. Specifically, we input the text

data into a transformer model, where they undergo multiple layers of self-attention mechanisms and feed-forward neural network layers, resulting in the output of the transformer model. Simultaneously, we use the syntactic structure graph as input, and apply the GNN model to iteratively update the nodes and edges of the graph to obtain the output of the GNN. Then, we use a cross-attention mechanism to fuse the output of the GNN with the linear layer output of the transformer model. To be more specific, we calculate the correlation scores between the GNN's output and the transformer linear layer output, and then use attention weights to perform weighted fusion. Finally, we reinput the result of the CAF to the linear layer of the transformer model for reprojection, combining the fused information with the original representation of the transformer model for downstream tasks. By employing this CAF method, we can fully leverage the transformer model's self-attention mechanism and the GNN model's ability to model syntactic structure graphs. This approach integrates the semantic information (transformer model) and the syntactic information (GNN model) of the text data, allowing their information to influence and complement each other using cross-attention fusion, thereby enhancing the model's ability to express and understand the text data. This CAF method is similar to the backend fusion approach in multimodal fusion, as it combines information from different data sources to improve the model performance. However, it differs from multimodal fusion in that it performs fusion between different representations of the same data source, rather than fusion across different modalities from different data sources.

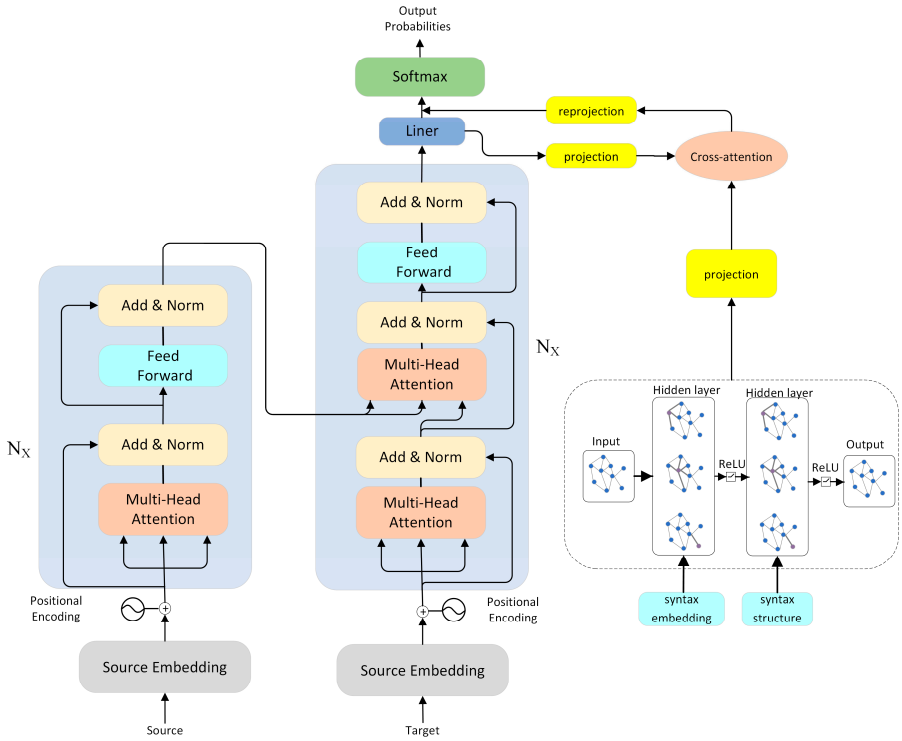

Figure 9. Model structure of cross-attention fusion.

4. Experiments

4.1. Data Preparation

To realize machine translation in the field of electrical engineering, this paper uses all Chinese and English parallel corpora from the electrical field as data sets. These corpora mainly come from some Chinese and English materials in the field of electrical engineering, including professional books [24,25], literature and some technical forums and official websites related to the electrical field to ensure the authority, professionalism and accuracy of the corpus. In the experiment, about 190,000 bilingual parallel corpora were used as the

training set, and 2000 bilingual parallel sentence pairs were used as the verification set and test set.

4.2. Experimental Setup and Method

This experiment uses the Ubuntu 18.04, a GPU processor, and machine translation training on a 2080 graphics card. This article utilizes the open-source pre-training model CLIP for pre-training (GitHub: https://github.com/openai/CLIP, accessed on 4 April 2023), while the baseline model employs the current mainstream transformer. The vocabularies of both Chinese and English are set to 40,000 words. Low-frequency words not included in this vocabulary are replaced with "UNK". Both languages employ Byte Pair Encoding (BPE) for word segmentation. The dimensions of word embedding and the hidden layer are both set to 512, while the dimension of the internal FFN (feed-forward neural network) layer is set to 2048. The model is trained using the Adam optimizer, and the learning rate is set to 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\varepsilon = 10^{-9}$. The model's random dropout rate is set at 0.1, and the batch size is 64. Beam search, with a search width of 5, is used for decoding.

We utilize the case-insensitive BLEU [26] value as the main automatic evaluation index to evaluate the translation quality based on *n-gram* accuracy calculation, as follows:

$$Count_{clip}(n\text{-}gram) = \min\{Count(n\text{-}gram), \text{MaxRe}fCount(n\text{-}gram)\} \tag{11}$$

$$P_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')} \tag{12}$$

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{13}$$

$$BLEU = BP \times \exp[(\sum_{n=1}^{N} w_n \log P_n)] \tag{14}$$

In the formula, *n-gram* represents the number of times that *n-gram* phrases appear in the machine translation results. In this paper, $n = 4$, r is the length of the reference translation and c is the length of the machine translation.

5. Result Analysis

5.1. Comparison of Fusion Methods

In this paper, we use the transformer [27] as the baseline model, and the model + CLIP pre-trained by the transformer and GNN is implemented first. The experimental results are shown in Table 1. Compared to the baseline model for the EN–ZH task, the translation model using CLIP pre-training shows an increase in the BLEU value of 3.37 percentage points, while the translation model using CAF shows an increase of 1.31 percentage points. Similarly, compared to the baseline model for the ZH–EN task, the translation model using CLIP pre-training shows an increase in the BLEU value of 3.18 percentage points, while the translation model using cross-attention fusion shows an increase of 1.03 percentage points. By comparing these three translation models, we can observe that the model pre-trained with CLIP achieves the best performance. Whether in the EN–ZH or ZH–EN translation tasks, the CLIP-pretrained translation model consistently outperforms the model with CAF by more than two percentage points in terms of the BLEU score. We hypothesize that the information obtained after CLIP pre-training can more fully reflect the syntactic and semantic information of the sentence.

Table 1. Experimental results of +CLIP and CAF models in EN–ZH and ZH–EN.

Model	EN–ZH		ZH–EN	
	BLEU	**Δ**	**BLEU**	**Δ**
Transformer [27]	34.54	-	29.23	-
+CLIP	37.91	↑ 3.37	32.41	↑ 3.18
CAF	35.85	↑ 1.31	30.26	↑ 1.03

Where: Δ is the BLEU score improved by the +CLIP model and the CAF model compared to the transformer model. ↑ represents "increase".

Analyzing the experimental results as shown in Table 1, we can draw the following conclusions:

(1) The +CLIP model outperforms the CAF model. We explore the reasons behind the superior performance of the +CLIP model by comparing its differences with the CAF model in the machine translation task:

   (a) Different data processing approaches: The CAF method utilizes a GNN to process syntactic tree graph data and employs a transformer model to handle textual data. Finally, a cross-attention mechanism is used to fuse them together. But, the +CLIP method leverages the CLIP model for multimodal pretraining, mapping both textual and visual data into the same multimodal feature space. This ability to fuse multimodal features enables the CLIP model to better understand and model the relationship between syntactic tree graphs and text, capturing the semantic information of sentences more effectively and thus achieving a better performance in machine translation tasks.

   (b) Different feature fusion approaches: CAF utilizes a cross-attention mechanism to fuse the syntactic tree graph and textual data. However, +CLIP achieves feature fusion using multimodal pretraining. This fusion approach maps the vector representations of text and images into the same multimodal feature space, allowing better utilization of this feature space containing information from both modalities during fine-tuning. The use of this shared feature space may contribute to improved generalization and representation capabilities, thereby enhancing the machine translation performance.

   (c) Semantic understanding capability: The +CLIP approach achieves cross-modal semantic understanding of text and images using multimodal pretraining, capturing the semantic correlations between them. However, the CAF method focuses on a semantic understanding of the sentence structure and dependencies. It utilizes a GNN to process syntactic tree graph data and employs a transformer model to handle textual data, aiming to better understand the structural composition and dependency relationships within sentences. While both methods possess semantic understanding capabilities, their emphasis and implementation differ, leading to variations in their performance.

These differences in feature fusion, semantic understanding and implementation contribute to the superior performance of the CLIP model:

(2) By comparing the translation results of the two EN–ZH and ZH–EN language pairs, it can be seen that the effect of syntactic information fusion is more obvious in EN–ZH. The main reason is that English is more complex in syntactic structure than Chinese. The syntactic structure of Chinese is mainly expressed based on the position and order of words, while English pays more attention to the expression of grammatical relationships, such as the relationship between subject and object, and the relationship between subordinate clauses and main clauses. Therefore, adding syntactic information to English translation can better capture the syntactic structure of the source language and help improve the translation accuracy and fluency.

Compared with the baseline systems in Table 1, it can be seen that integrating syntactic information into neural machine translation can indeed improve the translation accuracy

of the model, which also demonstrates the effectiveness of the two approaches proposed in this paper for incorporating syntactic information into neural machine translation models.

5.2. Comparison of BLEU Values of Different Sentence Lengths

We assume that longer sentences contain more information, and we divide sentence length into six groups by intervals of 10. We compare the baseline model, +CLIP and CAF, using the BLEU scores to evaluate the translation quality across different sentence lengths. The results are presented in Figures 10 and 11.

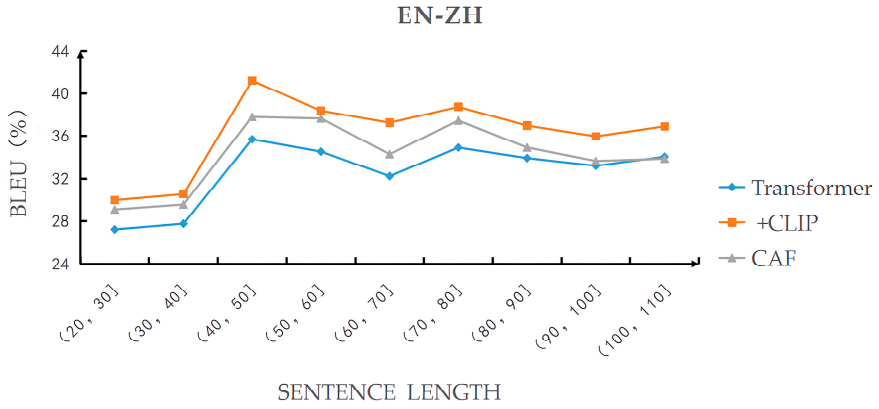

Figure 10. Comparison of BLEU values of different sentence lengths on EN–ZH.

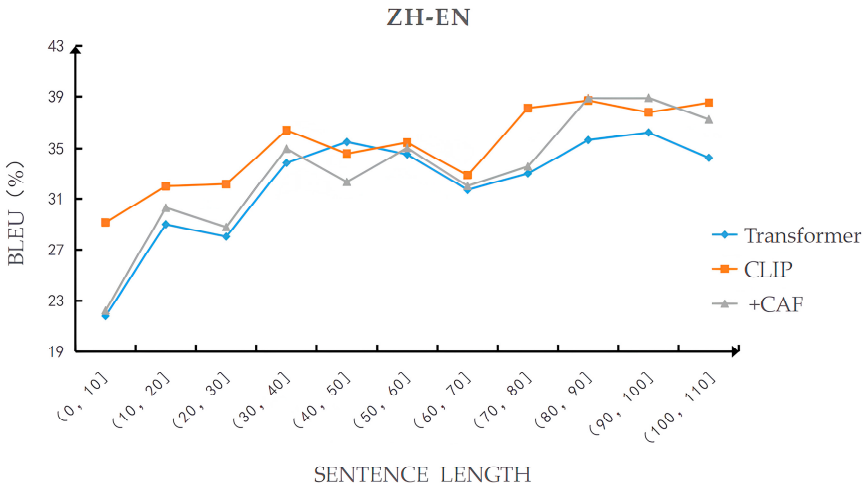

Figure 11. Comparison of BLEU values of different sentence lengths on ZH–EN.

Figures 10 and 11 demonstrate that the two methods proposed in this paper achieve better translation results than the baseline model for both the EN–ZH and ZH–EN tasks. Analyzing the experimental results reveals that syntactic information plays a crucial role in machine translation, particularly for longer and complex sentences. It provides the model with additional contextual information and grammar rules, aiding in a better understanding of sentence structure, resolving ambiguities and polysemy and guiding the model to generate more accurate and grammatically appropriate translations that align with the target language's rules and conventions.

5.3. Ablation Experiment

To explore the specific effects of the two proposed methods, two ablation experiments are designed in this paper. The first is to retain the fusion method of cross-attention unchanged, and compare the effect of using CLIP pre-trained data on the translation results. The second is to use CLIP pre-trained data by comparing the effect of simple average fu-

sion and our cross-attention fusion method on the translation results. The principle of the average fusion method is to simply average the output of the transformer and the output of the GNN to predict the probability distribution of the target vocabulary. The model structure diagram of average fusion is similar to Figure 9, just replacing the cross-attention part with the average fusion function. The two comparison models in the first ablation experiment are AF + CLIP and CAF. The former uses both CAF and +CLIP, while the latter only uses CAF without +CLIP. The two comparison models in the second ablation experiment are CAF + CLIP and AF + CLIP. The former utilizes CAF and +CLIP, and the latter utilizes average fusion and +CLIP. The experimental results are shown in Table 2.

Table 2. Experimental results of two ablation experiments.

Model	EN-ZH		ZH-EN	
	BLEU	**Δ**	**BLEU**	**Δ**
CAF	35.85	-	30.98	-
CAF + CLIP	31.87	↓ 3.98	25.35	↓ 5.63
AF + CLIP	30.89	↓ 5.05	24.75	↓ 6.23

Where: Δ is the BLEU score reduced by the CAF + CLIP model and the AF + CLIP model compared to the CAF model. ↓ stands for "lower".

By analyzing the experimental results shown in Table 2, the following can be obtained:

(1) The translation model CAF + CLIP that uses both cross-attention fusion and CLIP pre-training cannot effectively improve the performance of the model, but will cause a significant decline in the quality of the translation. This paper argues that the reason for this result is that the characteristics of CAF and +CLIP are not suitable for simultaneous use. The core of CAF is to fuse syntactic tree graph data and text data to improve the translation ability of the model. Its main advantage is that it can use the relationship between words in the syntactic tree graph data to improve the model's ability to understand the semantics of sentences. However, this approach may be replaced by CLIP pre-trained models that have learned rich semantic representations from multiple vision and language tasks using large-scale self-supervised learning. Using CAF and CLIP pre-trained models at the same time may lead to redundant information fusion, which affects the performance of the model. In addition, the CAF + CLIP model needs to process a large number of parameters during training, which may cause overfitting and increase the training time, thus affecting the performance of the model. Therefore, models pre-trained using cross-attention fusion or CLIP alone can better utilize their respective advantages and improve the performance of the model.

(2) After using CLIP pre-training, the difference between the translation model using cross-attention fusion and the translation model using average fusion is not significant. However, in general, the translation model with cross-attention fusion performs better. The reason may be that cross-attention fusion can better utilize the information interaction between different models, thus improving the representation ability of the model. In the simple average model, the information of different models is simply averaged and fused, and the interactive information between different models cannot be fully utilized. In cross-attention fusion, each model can better understand the information of other models by paying attention to other models, thereby avoiding information conflicts between different models and improving the model performance. CAF can also avoid information conflicts between different models. In simple average models, information from different models may collide, resulting in poor model performance.

To sum up, when performing text translation tasks, it is necessary to select an appropriate model structure and pre-training technology according to the characteristics and requirements of the task to improve the translation quality. At the same time, it is necessary to pay attention to the interaction between different model structures and pre-training

techniques to avoid excessive introduction of redundant information, which will lead to a decrease in translation quality.

6. Conclusions

This paper proposes two effective methods for integrating source-side syntactic information into neural machine translation models. The first method is based on CLIP pre-training technology, which utilizes advanced natural language processing techniques to extract both text and syntactic tree information and propagate them to a transformer model to complete machine translation tasks. In English–Chinese and Chinese–English translation tasks, this method results in BLEU score improvements of 3.37 and 3.18 percentage points, respectively. The second method is the adoption of cross-attention fusion, which can better capture the relationship between the vocabulary in the syntactic tree and improve the model's understanding of the sentence semantics. In English–Chinese and Chinese–English translation tasks, this method results in BLEU score improvements of 1.31 and 1.03 percentage points, respectively. The paper performed ablation experiments by integrating the two proposed models, but unfortunately, the results were unsatisfactory. Furthermore, this paper also investigates the impact of different sentence lengths on the translation results, providing important reference for researchers. These research results are of great significance in advancing machine translation technology and improving the quality of machine translation models.

However, this study has certain limitations. Specifically, it is confined to English-to-Chinese and Chinese-to-English translation tasks, necessitating further research on translation tasks involving other languages. Additionally, the study solely focuses on integrating source-side syntactic information and does not address how to incorporate target-side syntactic information into the translation model. In translation tasks, target-side syntactic information may also have an impact on the translation results, requiring further investigation. In the future, our research direction will focus on incorporating rich syntactic information of the target language into neural machine translation models and further expanding the experimental scope to cover more languages. Additionally, we will explore how to integrate syntactic analysis of multimodal information such as visual and audio data to enhance the performance of our models in multilingual translation tasks. The goal of this research direction is to improve the translation quality and accuracy of machine translation models by leveraging the complementarity of syntactic information and multimodal data, in order to meet diverse language translation needs.

Author Contributions: Writing—original draft, Y.S.; writing—review and editing, Y.C. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the National Natural Science Foundation of China (grant no. U2004163).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to being part of an ongoing research project.

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

References

1. Tiejun, Z. *Principles of Machine Translation*; Harbin Institute of Technology Press: Harbin, China, 2001.
2. Zhiwei, F. Machine Translation: From Rule-Based Techniques to Statistics-Based Techniques. In Proceedings of the 2010 China Translation Professional Exchange Conference Proceedings, Macau, China, 6–8 November 2010.
3. Minghu, G.; Zhiqiang, Y. A Survey of Neural Machine Translation. *J. Yunnan Univ. Natl. Nat. Sci. Ed.* **2019**, *28*, 5.
4. Chen, H.; Huang, S.; Chiang, D.; Chen, J. Improved Neural Machine Translation with a Syntax-Aware Encoder and Decoder. *arXiv* **2017**, arXiv:1707.05436.

5. Aharoni, R.; Goldberg, Y. Towards String-to-Tree Neural Machine Translation. *arXiv* **2017**, arXiv:1704.04743.
6. Zheng, X.; Chen, H.; Ma, Y.; Wang, Q. Neural machine translation model combining dependency syntax and LSTM. In *ITM Web of Conferences*; EDP Sciences: Les Ulis, France, 2022.
7. Duan, S.; Zhao, H.; Zhou, J.; Wang, R. Syntax-aware Transformer Encoder for Neural Machine Translation. In Proceedings of the International Conference on Asian Language Processing (IALP), Shanghai, China, 15–17 November 2019.
8. Ma, C.; Tamura, A.; Utiyama, M.; Sumita, E.; Zhao, T. Improving Neural Machine Translation with Neural Syntactic Distance. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019.
9. Bugliarello, E.; Okazaki, N. Enhancing Machine Translation with Dependency-Aware Self-Attention. *arXiv* **2019**, arXiv:1909.03149.
10. Gong, L.; Guo, J.; Yu, Z. Neural Machine Translation Method Based on Source Language Syntax Enhanced Decoding. *J. Comput. Appl.* **2022**, *42*, 3386.
11. Gao, S.; Zhang, Y.; Yu, Z.; Zhu, H. *Chinese-Vietnamese Parallel Sentence Extraction Method with Syntactic Structure and Tree-LSTM Integration:CN202010978713.0[P]*; ResearchGate: Berlin, Germany, 2021.
12. Wang, Z.; He, J.; Yu, Z.; Wen, Y.; Guo, J.; Gao, S. Chinese-Vietnamese Convolutional Neural Machine Translation with Syntactic Parsing Trees Integration. *J. Softw.* **2020**, *31*, 11.
13. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021.
14. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
15. Du, Q. *A Multimodal Machine Translation Data Augmentation Method Based on Image Description: CN202011212067.3[P]*; ResearchGate: Berlin, Germany, 2021.
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
17. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–41. [CrossRef]
18. Zheng, Y.; Gindra, R.H.; Green, E.J.; Burks, E.J.; Betke, M.; Beane, J.E.; Kolachalama, V.B. A graph-transformer for whole slide image classification. *IEEE Trans. Med. Imaging* **2022**, *41*, 3003–3015. [CrossRef] [PubMed]
19. Shi, L.; Ji, Q.; Chen, Q.; Zhao, H. A Review of the Application of Visual Transformers in Medical Image Analysis. *Comput. Eng. Appl.* **2023**, *59*, 41–55.
20. Xie, F.; Zhang, D.; Liu, C. Global–Local Self-Attention Based Transformer for Speaker Verification. *Appl. Sci.* **2022**, *12*, 10154. [CrossRef]
21. Yang, L.; Guo, W.; Han, F. Chinese Speech Recognition Based on DFCNN-CTC and Transformer. *Fire Control Command. Control* **2022**, *47*, 6.
22. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, S.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
23. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 27730–27744.
24. Fan, Y. A Study on Chinese-English Machine Translation Based on Migration Learning and Neural Networks. *Int. J. Artif. Intell. Tools* **2022**, *31*, 2250031.
25. Wang, H. Short Sequence Chinese-English Machine Translation Based on Generative Adversarial Networks of Emotion. *Comput. Intell. Neurosci.* **2022**, *2022*, 3385477. [CrossRef] [PubMed]
26. 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.
27. 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:1910.03771.