electronics

Article

Ancient Text Translation Model Optimized with GujiBERT and Entropy-SkipBERT

Fuxing Yu [1,2], Rui Han [1], Yanchao Zhang [3] and Yang Han [4,*]

[1] School of Artificial Intelligence, North China University of Science and Technology, Tangshan 063210, China; yfx626@ncst.edu.cn (F.Y.); hanr@stu.ncst.edu.cn (R.H.)
[2] Hebei Key Laboratory of Industrial Intelligent Perception, North China University of Science and Technology, Tangshan 063210, China
[3] Department of Resource and Environmental Engineering, Hebei Vocational University of Technology and Engineering, Xingtai 054000, China; xkb@ncst.edu.cn
[4] School of Science, North China University of Science and Technology, Tangshan 063210, China
* Correspondence: hanyang@ncst.edu.cn

Abstract: To cope with the challenges posed by the complex linguistic structure and lexical polysemy in ancient texts, this study proposes a two-stage translation model. First, we combine GujiBERT, GCN, and LSTM to categorize ancient texts into historical and non-historical categories. This categorization lays the foundation for the subsequent translation task. To improve the efficiency of word vector generation and reduce the limitations of the traditional Word2Vec model, we integrated the entropy weight method in the hopping lattice training process and spliced the word vectors with GujiBERT. This improved method improves the efficiency of word vector generation and enhances the model's ability to accurately represent lexical polysemy and grammatical structure in ancient documents through dependency weighting. In training the translation model, we used a different dataset for each text category, significantly improving the translation accuracy. Experimental results show that our categorization model improves the accuracy by 5% compared to GujiBERT. In contrast, the Entropy-SkipBERT improves the BLEU scores by 0.7 and 0.4 on historical and non-historical datasets. Ultimately, the proposed two-stage model improves the BLEU scores by 2.7 over the baseline model.

Keywords: Chinese ancient texts; GujiBERT; machine translation; text classification; dependency analysis; SkipGram

check for updates

Citation: Yu, F.; Han, R.; Zhang, Y.; Han, Y. Ancient Text Translation Model Optimized with GujiBERT and Entropy-SkipBERT. *Electronics* **2024**, *13*, 4492. https://doi.org/10.3390/electronics13224492

Received: 22 October 2024
Revised: 12 November 2024
Accepted: 13 November 2024
Published: 15 November 2024

1. Introduction

In recent years, machine translation technology [1] has developed rapidly, primarily driven by neural network models, resulting in a significant improvement in the translation quality of modern languages. However, ancient language translation still faces unique challenges, mainly due to differences in language structure and grammatical features compared to modern languages. The syntactic structure of ancient languages was more flexible, and the word order varied. For example, in ancient Chinese, the subject–verb object order is relatively free, and even some components are often omitted, while modern language usually follows a more fixed sentence structure. This flexibility brings parsing difficulties to traditional machine translation models. The phenomenon of polysemy is common in the ancient Chinese literature. The same word can have multiple meanings in different contexts, and machine translation makes it difficult to accurately determine its meaning based solely on context, leading to ambiguity. Modern language has relatively straightforward word meanings, reducing the possibility of translation ambiguity. The classical Chinese literature contains many vocabulary and allusions that rely on specific cultural backgrounds, and understanding these contents requires relevant historical and cultural knowledge. In contrast, modern language expression is more direct and explicit, with lower cultural dependence. Classical Chinese lacks a rich bilingual parallel corpus, while modern languages have sufficient data resources to support machine translation training. The lack of this corpus limits the effectiveness of machine translation models in translating ancient texts.

To further improve the accuracy of ancient text translation, this paper proposes a two-stage text translation model, which aims to capture the semantics of ancient texts more accurately by combining multiple deep learning techniques. The model first classifies ancient texts through a structure that combines GujiBERT [2], Graph Convolutional Networks (GCNs) [3], and Long Short-Term Memory Networks (LSTMs) [4], which classify ancient texts into historical and non-historical categories based on the type of document. The purpose of this classification is to lay the foundation for the subsequent translation task because different types of ancient texts have significant differences in content and language style, and the post-classification processing helps to improve the relevance of the model and the translation effect.

On this basis, this paper proposes an improved word vector generation method for the existing Word2Vec [5], which applies the entropy weighting method to the training process of SkipGram [6] and combines it with GujiBERT to optimize the word vector generation process. The entropy weighting method [7] is a statistical method commonly used for weight allocation, and the importance of words in different contexts can be determined more objectively by calculating the entropy value. Introducing the entropy weighting method into the training of word vectors can effectively enhance the model's ability to capture lexical polysemy and complex grammatical structures in ancient texts, thus improving the semantic comprehension ability of the translation model.

The main contributions of this work are as follows.

(1) We propose a classification model for ancient texts combining GujiBERT, GCN, and LSTM, laying the foundation for translating different ancient texts.
(2) We design a word vector training method that introduces the entropy weight method in the training of the SkipGram model, optimizes the generation process of word vectors, and splices them with the word vectors obtained from the GujiBERT training to improve the model's ability to capture lexical polysemy and complex grammatical structures in ancient texts.
(3) Our proposed method trains the model separately for different types of ancient text datasets and achieves the accurate translation of historical and non-historical texts.

2. Related Works

With the development of machine translation, attempts to combine classification models with machine translation have begun to solve the language changes brought by different domains. Classification models help translation systems adaptively adjust translation strategies by recognizing features such as the text's domain, style, or context, thus improving translation quality. Li [8] proposed a two-stage sentiment quaternion extraction framework based on machine translation and text categorization. The first stage treats the sentiment quaternion extraction task as machine translation and realizes the extraction of aspect–opinion pairs through a span-based labeling scheme and a question-and-answer mechanism. The second stage treats the categorization of aspect categories and sentiment polarity as a text-generation task, utilizing natural language generation to enhance the semantic representation of sentiment elements. Finally, the results of the two phases are merged through a template generator to decode the complete sentiment quaternion. Although the method has improved emotion quaternion extraction, some shortcomings remain. The difference between tree structure and sequence information is significant. The technique of fusing tree structure information with virtual template words has a limited effect on the comprehensive utilization of structure information. It is difficult to bridge the difference between tree structure and sequence information completely. Wu et al. [9] divided ancient texts into three periods, ancient, middle-aged, and near-ancient, according to the characteristics of the ancient texts, and used the corpus of each period to train various machine translation models. The experimental study shows that the machine translation model training of ancient texts divided into three periods can improve the accuracy and fluency of the ancient translation models. Still, the deep semantic analysis and understanding of the characteristics of each period are not yet sufficient, especially in the rhetorical devices, allusions, and other aspects that have not been fully emphasized.

According to existing research, introducing classification models in machine translation to classify text according to its features can significantly improve the accuracy and naturalness of translation. In recent years, pre-trained models have performed well in text classification, and by pre-training on large-scale text data, the pre-trained models can capture deep linguistic features, making text classification more accurate. Despite the excellent performance of BERT variants in modern language processing, they have certain limitations in the task of ancient text translation. This limitation mainly stems from the differences in syntactic structure, vocabulary usage, and contextual understanding between ancient and modern texts, and traditional pre-training models are too complex to meet these specific needs adequately. Liu et al. [10] proposed the RoBERTa model, which achieved significant results through training and optimization on large-scale data. Still, due to the sparse contextual information and concise syntactic structure of the ancient text corpus, it is difficult to capture the deep linguistic features of ancient texts effectively. Sanh et al. [11] proposed the DistilBERT model, which improves computational efficiency by reducing the number of parameters. Although it performs well in resource-constrained environments, its performance is still insufficient for ancient texts' complex syntactic structure and deep cultural background. Cui et al. [12] proposed the BERT-wwm model. This model enhances the comprehension of modern Chinese. Still, it is mainly trained on the modern Chinese corpus, which makes it difficult to cope with the significant differences between ancient texts and modern Chinese in terms of lexical and semantic features. Wang et al. [2] proposed the GujiBERT model, which significantly improves the performance in comprehension tasks such as automatic sentence breakage, linguistic annotation, and entity recognition through self-supervised training on the Simplified-Traditional Chinese (STCW) ancient text dataset. Performance: Compared to previous small-scale models, GujiBERT excels in fine-grained processing tasks such as sentence breaking and lexical annotation but has limited capability in generative tasks and needs to be combined with other generative models. In addition, compared to larger-scale models, GujiBERT has a smaller parameter size, which may limit its performance in processing complex tasks.

In the study of word embedding models, traditional methods such as Word2Vec have significant results in capturing semantic relations between words. However, they are still deficient in dealing with complex dependency and polysemous words. Based on CBOW, Zheng et al. [13] obtain word vectors based on dependency words and context word prediction targets and combine the obtained word vectors with LSTM as the input sequence. This method effectively improves the performance of the translation model, but the translation accuracy decreases when dealing with longer texts. Xin et al. [14] conducted a comparative study of two models of Word2Vec, SkipGram, and CBOW, and the experimental results showed that the SkipGram model has a more obvious advantage in neologism recognition when Chinese word vectors are trained through a large corpus.

Existing studies have improved the translation adaptability of domain-specific texts by combining classification models with machine translation. However, they still face some limitations, such as the insufficiency of complex dependency modeling, the processing of deep semantic and cultural elements to be further optimized, and the insufficient performance of the generative task capability and the translation of long texts. To solve these problems, the modeling of complex dependencies should be further optimized to enhance the deep-level understanding of the text and cultural adaptation. More effective text categorization and translation methods are explored to improve the model's capture of complex dependencies and deep semantic features.

3. Methods

3.1. Phase One: GujiBERT-GCN-LSTM Model for Classical Chinese Text Classification

Ancient texts are significantly different from modern texts due to the linguistic features, structures, and contexts of ancient texts, including the change in meaning of the words used, the differences in syntactic structures, and the uniqueness of rhetorical techniques, which leads to the poor performance of traditional text classification models in classifying ancient texts. To address the above problems, a GujiBERT-GCN-LSTM-based text classification model for ancient texts is proposed, which efficiently processes the complex structure and semantics of ancient texts by closely integrating a series of modules. The model structure of the GujiBERT-GCN-LSTM model is visualized in Figure 1.

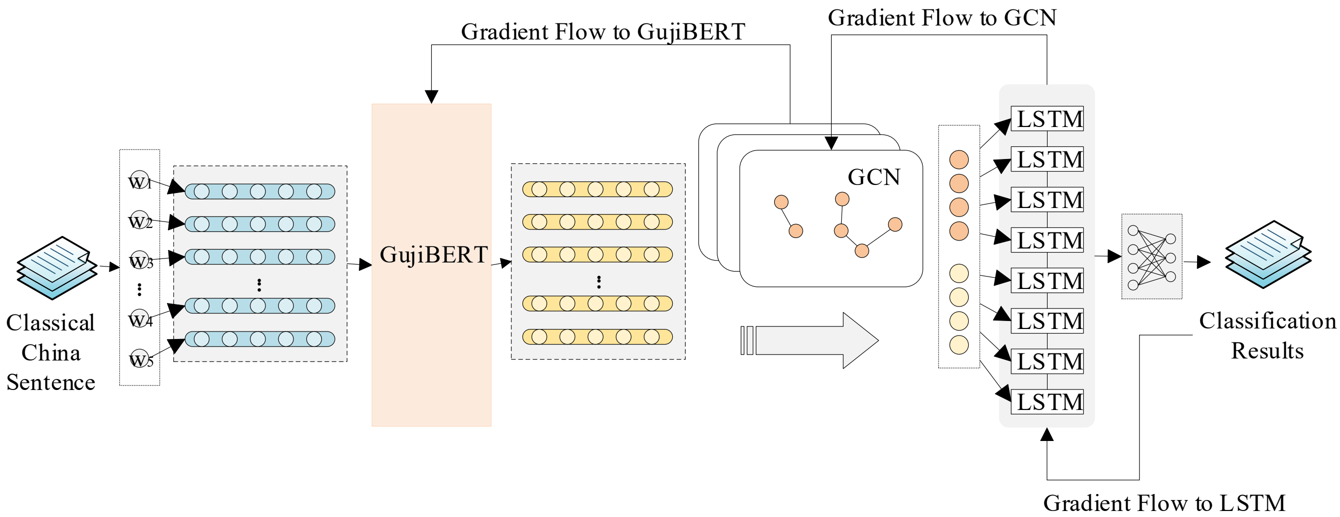

Figure 1. Structure of GujiBERT-GCN-LSTM model.

Firstly, the model takes the ancient text as input and afterword segmentation and encoding, and it is fed into the GujiBERT model. GujiBERT is based on the Transformer architecture, which has powerful bidirectional encoding capability and captures the semantic information of each word in context through the self-attention mechanism. It first transforms the text into word embeddings and positional embeddings. Then, it analyses the contextual relationships between words through a multi-layer Transformer encoder, thus outputting the contextual embedding vectors for each word. These embedding vectors contain the semantic understanding of each word in the text in its context and are the basis for subsequent processing.

The GCN module is used to capture nonlinear word relationships that are unique to ancient texts. Although GujiBERT can capture contextual dependencies, word relationships in ancient texts often go beyond linear order, such as inversions or metaphors. GCN constructs a graph structure by treating each word as a node, and the edges represent the dependencies between words. The representation of a node is updated by aggregating the features of its neighboring nodes, and this process is unfolded by a multilayer convolution operation so that each node not only contains its information but also integrates the features of its neighboring nodes. GCN performs a feature transformation with the following formula:

$$H^{(l+1)} = \sigma\left(\hat{A}H^{(l)}W^{(l)}\right) \tag{1}$$

In Equation (1), $H^{(l)}$ is the layer l node characteristic matrix, $W^{(l)}$ is the learnable weight matrix of the layer l, and σ is the activation function ReLU.

After GujiBERT's contextual understanding, GCN's complex relationship capture, and LSTM's long-distance dependency processing, the model generates a final representation vector for each word. To classify the text, the model generates the overall text representation by average pooling and then classifies the text by fully connected layers and the Softmax function. Ultimately, the error between the predictions and the true labels is calculated using the cross-entropy loss function, and the model is optimized using backpropagation, constantly updating the parameters in GujiBERT, GCN, and LSTM. In this way, the model is able to comprehensively understand ancient texts from multiple levels of semantics, structure, and dependencies, thus achieving more accurate text classification.

3.2. Phase Two: Improved Entropy-SkipBERT Model for Classical Chinese

A dependency analysis refers to obtaining the dependency relationship between words in a sentence, which is a directed unequal relationship between a central word and its related words, in which the core word dominates its related words, and the related words depend on the core word. The open-source Natural Language Processing (NLP) library SpaCy (https://spacy.io (accessed on 7 November 2024)) provides dependency-parsing capabilities, enabling a syntactic analysis of sentences.

The dependency syntactic analysis of ancient texts using SpaCy to generate dependency syntactic trees is complicated because there are 22 common types of relations, and more types of relations may lead to the problem of model overfitting. Therefore, the dependency relations were filtered after the annotation was completed, and only eight types of dependency relations were retained: the subject–predicate, verb–object, definite–medium, parallel, punctuation, modifier, dependency marker, and compound structure.

SkipGram is a Word2Vec model for generating word embedding architecture from text. Compared with CBOW [15], another architecture in Word2Vec, SkipGram handles rare words better. As opposed to CBOW, which predicts the central word given the context, SkipGram is a selected central node that predicts the surrounding context nodes, and through the context node's Conditional Probability Learning word vectors, calculating the conditional probability of the target word given the context word-specific formula is as shown in Formula (2), where the target word is w_O, given the context word w_I; v_{w_I} is the vector representation of the input word; v_{w_O} is the vector representation of the output word; and W is the total number of words in the vocabulary.

$$P(w_o \mid w_I) = \frac{\exp\left(v_{w_o}^T v_{w_I}\right)}{\sum\limits_{w=1}^{W} \exp(v_w^T v_{w_I})} \tag{2}$$

The SkipGram model trains word vectors by maximizing conditional probability, typically producing more accurate and enriched word embeddings during training. The SkipGram model processes each target context separately, enabling it to capture complex lexical relationships better. The structure of the SkipGram is shown in Figure 2. In Figure 2, taking a sentence of eight words as an example, each word is denoted as w. Selecting $w(3)$ as the center word, it serves as the input for the SkipGram model. This center word is mapped to a word vector through the projection layer, embedding the semantic information of the word. Using the word vector of the center word, the model predicts its context words, specifically $w(1)$, $w(2)$, $w(4)$, and $w(5)$. The SkipGram model optimizes the word vectors by maximizing the conditional probability of the context words given the center word, enabling the vectors to represent the relationships between words better.

When training a neural network, the weights are adjusted with training. Therefore, the computational size of the weight matrix during the training of the SkipGram model will be significant, consuming a large amount of computational resources and slowing down the training speed. To address this problem, the negative sampling technique [16] is used to optimize the training process. Negative sampling enhances training efficiency and reduces computational complexity by updating only a subset of weights. This is carried out by incorporating a few negative samples along with the positive ones and updating only the selected samples rather than the entire word list.

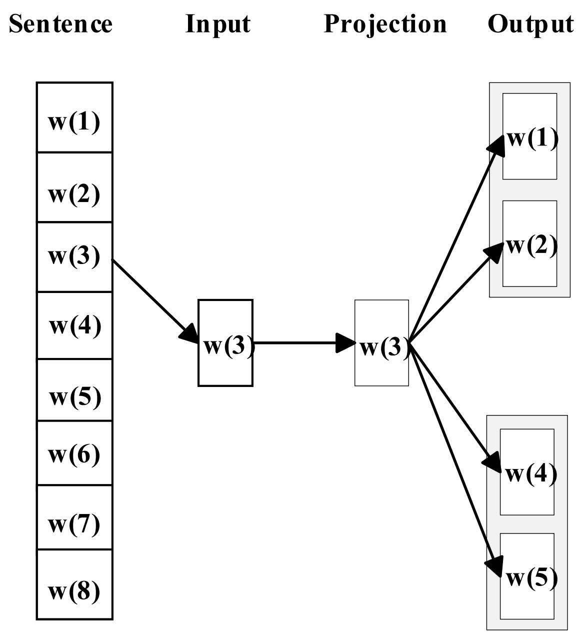

Figure 2. Structure of SkipGram model.

In this paper, we enhance the SkipGram model by incorporating dependency weights and distance factors to refine the probability distribution of context words. The dependency relationships are processed using SpaCy, and we implement an entropy weighting method to assign these weights based on the identified dependencies. This entropy weighting approach allocates weights objectively by analyzing the information associated with each indicator. Specifically, a low information entropy for an indicator suggests more significant variability across different contexts, warranting a relatively high weight. Conversely, a high information entropy indicates less variability, leading to a lower weight assignment.

Initially, we conduct a dependency analysis of the ancient texts, calculating the frequency of each dependency type. These frequencies are normalized such that their total sums are one. The formula for calculating the frequency of each dependency is as follows:

$$p_i = \frac{f_i}{N} \tag{3}$$

where f_i is the number of occurrences of dependency d_i in the dataset and N is the total number of all dependencies in the dataset.

The uncertainty of the dependencies is measured by calculating the entropy value, which is calculated as follows:

$$H = -\sum_i p_i \log(p_i) \tag{4}$$

where p_i is the frequency of the ith dependency.

After obtaining the entropy value, the weight of each dependency is calculated. The higher the weight, the greater the importance of the dependency in the whole sentence structure, and its weight is calculated by the formula

$$w_i = \frac{1}{1 - H} \tag{5}$$

The weights obtained are normalized to achieve the weights, and the formula is specified as follows:

$$w'_i = \frac{w_i}{\sum_j w_j} \tag{6}$$

Dependency weights were calculated for historical texts, as shown in Table 1, and for non-historical texts, as shown in Table 2.

Table 1. Historical category dependency weights.

Dependency Relationship	Weights
punct	0.0199
dobj	0.0467
conj	0.0618
advmod	0.0616
nsubj	0.0606
amod	0.1649
acl	0.2462
nummod	0.3383

Table 2. Non-historical category dependency weights.

Dependency Relationship	Weights
punct	0.0492
dobj	0.0543
conj	0.0813
advmod	0.0662
nsubj	0.0559
amod	0.1519
acl	0.2119
nummod	0.3293

Dependency weights are introduced into the SkipGram model so that the stronger the dependency with the central word in the sentence, the higher the probability of being generated. The Entropy-SkipBERT conditional probability formula is as follows: where w_t is the central word, w_c is the context word; v_{w_t} and v_{w_c} are the word vectors of the central word and the context word, respectively; V is the vocabulary list; and β is the weights of the dependency.

$$P(w_c \mid w_t) = \frac{e^{\sigma(\beta_r)(v_{w_c} \cdot v_{w_t})}}{\sum\limits_{w' \in V} e^{\sigma(\beta_r)(v_{w'} \cdot v_{w_t})}} \tag{7}$$

In this paper, the SkipGram model that introduces the entropy weighting method is referred to as Entropy-SkipGram. Entropy-SkipGram and GujiBERT are two different word vector training models. GujiBERT is a language model based on the Transformer architecture, which performs large-scale, unsupervised training through the mechanism of self-attention and can capture more complex semantic and contextual dependencies. The word vectors generated by GujiBERT usually have higher dimensionality, which can cover deeper semantic information. After Entropy-SkipGram and GujiBERT have been trained separately, they can be directly spliced and fused into a new word vector. Entropy-SkipGram word vectors and GujiBERT word vectors of the same word are spliced in vector dimensions to form a higher dimensional word vector. We refer to the overall module as Entropy-SkipBERT. Through this direct splicing operation, it is possible to retain the features and semantic information of the two models simultaneously, forming a richer representation of the word vector. This newly generated word vector is used as the input embedding layer. When receiving this fused word vector, the machine translation model can take advantage of the local contextual information captured by Entropy-SkipGram and the global semantic representation generated by GujiBERT, thus enhancing the translation effect. This combination of multi-source information makes the model more comprehensive and precise when dealing with word meanings, context dependencies, and sentence structure.

3.3. The Overall Framework of the Two-Stage Ancient Language Translation Model

The overall structure of the two-stage model is shown in Figure 3.

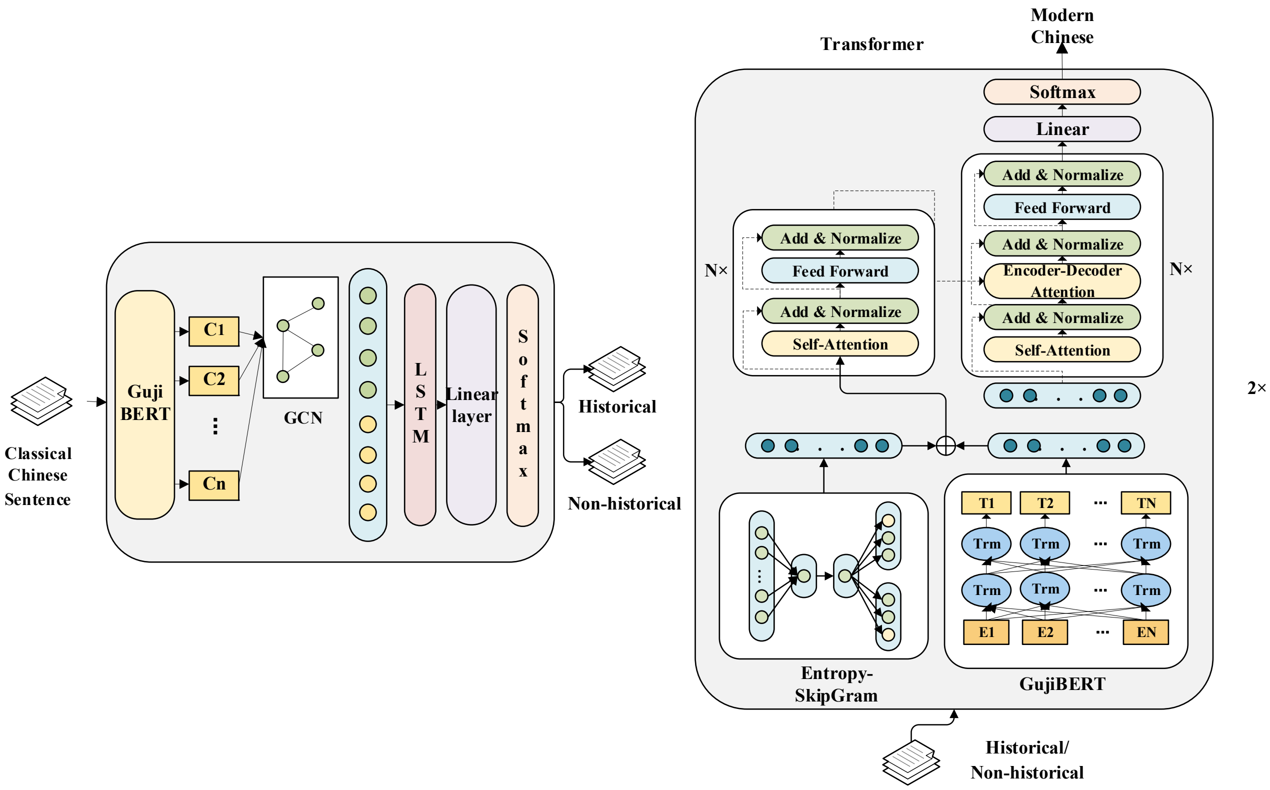

Figure 3. Structure of two-stage ancient text translation model.

The overall process of the two-stage ancient text translation model mainly consists of two stages: ancient text classification and ancient text translation. The first stage is the ancient text categorization stage. The ancient text is input into GujiBERT-GCN-LSTM to obtain two categories of text files translated: the historical class and non-historical class. The second stage is the ancient text translation stage. The history and non-history class datasets are machines translated using their respective Transformer model as the basis. Since the language styles and expressions of the historical and non-historical texts may differ significantly, training the Transformer translation model separately for each text class can improve translation adaptability and accuracy. To further enhance the word representation ability of the model, Entropy-SkipBERT is introduced into the translation model as a word vector training model to improve the semantic representation of word vectors. Documents from historical and non-historical texts are inputted into the respective trained Transformer translation models to generate the corresponding translations into modern Chinese. Through this categorized translation strategy, the model can process different types of texts more accurately so that the historical texts can retain their unique style and context. In contrast, the non-historical texts can be more closely adapted to modern Chinese expression habits. Such a phased approach makes the model more effective in generating fluent and accurate translations while considering the characteristics of ancient texts.

4. Experiments

4.1. Dataset

The dataset used for the experiment is from a publicly available dataset, Classical-Modern (https://github.com/NiuTrans/Classical-Modern (accessed on 9 November 2024)). Due to the small amount of non-historical literature data, there is more historical data than non-historical data, which can lead to bias in the model's text classification. The specific manifestation is that the recall rate is high, but the accuracy is low. The model tends to predict the input text as historical, thereby improving the recall rate of recognizing historical text. However, due to insufficient non-historical data samples, the model finds it difficult to effectively learn the features that distinguish between the two types of data, resulting in an increased misjudgment rate and a decrease in accuracy when classifying non-historical text. Although the model can comprehensively recognize historical texts, it is easy to misclassify non-historical texts as historical. When training machine translation models for non-historical texts, the model cannot thoroughly learn the features of non-historical texts during training, resulting in unstable translation performance. This makes it difficult for the model to generalize widely to non-historical texts, resulting in poor performance in practical applications when faced with unseen non-historical expressions. To solve this problem, a data enhancement technique is used to enhance the data of non-historical documents. The data augmentation process involves synonym replacement and incorporates a selection of bilingual ancient poems from the open-source Chinese Poetry dataset (https://github.com/chinese-poetry/chinese-poetry (accessed on 9 November 2024)). When enhancing synonym data for non-historical texts in ancient texts, the first step is to establish a synonym library, replace synonyms through context-based automatic replacement functions, and use SpaCy to perform a dependency analysis on sentences to ensure that the syntactic structure remains unchanged after replacement. By combining the manual review and optimization of the generated results, we ensure that the data quality conforms to ancient texts' style and semantic consistency, providing richer and more authentic training data for ancient text machine translation models.

The preprocessing of the bilingual dataset mainly includes corpus cleaning, word splitting, and building word lists. Corpus cleaning mainly includes three operations, namely, removing duplicated data, removing empty lines, and removing bilingual pairs containing book titles. Most modern texts containing book titles reference other allusions to translate ancient texts, which will impact model learning and the establishment of word lists. Therefore, the bilingual pairs containing book titles need to be removed. Segmentation is divided into ancient text segmentation and modern text segmentation. Ancient texts are usually simple in style; each word contains important semantic information, and ancient texts have the characteristics of multiple meanings of words. Therefore, word-by-word segmentation is used for ancient text segmentation, while BPE [17] is used for modern text segmentation. The dataset is split into two sections, one for training the classification model and the other for the translation model. The results of the data division are shown in Table 3.

Table 3. Results of the segmentation of the dataset from the classification model.

Datasets	Number of Sentences
Historical training set	14,400
Non-historical training set	14,400
Historical validation set	1800
Non-historical validation set	1800
Historical test set	1800
Non-historical test set	1800

The results of the datasets used to train the translation model division are shown in Table 4.

Table 4. Results of the dataset division.

Datasets	Number of Sentences
Training	21,600
Validation	2700
Test	2700

4.2. Experimental Parameter Setting

All of the models in this article are implemented using the Pytorch 2.3.1+cu118 framework. The learning rate of the classification model is 2×10^{-5}, the maximum sequence length is 38, and the hidden layer size is 768. For the SkipGram model, the word embedding dimension is 150, the window size is 5, and the learning rate is 0.01. The translation model uses the OpenNMT [18] toolkit with a learning rate of 0.1, 4 layers each for the encoder and decoder, and 8 multi-attention heads.

4.3. Experimental Results of the Classification Model

The classification model evaluation metrics are accuracy, recall, and the F1 score. Figure 4 shows the confusion matrix for the classification results.

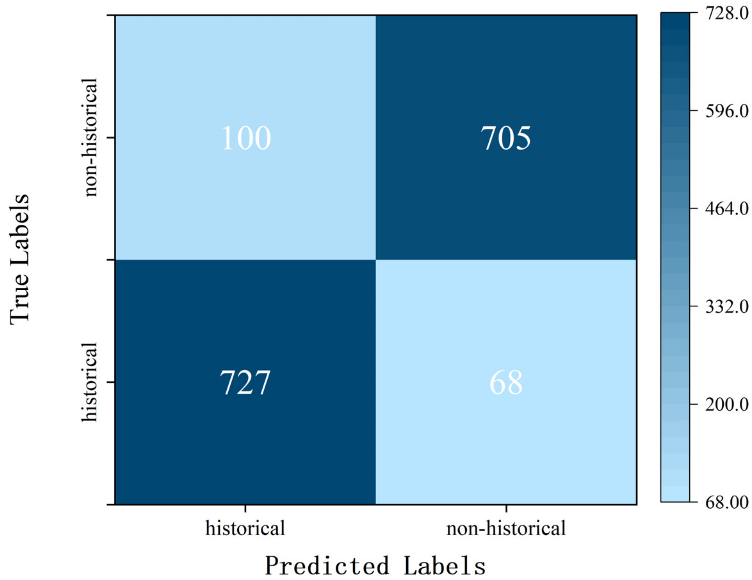

Figure 4. Classification model confusion matrix.

Figure 4 shows the confusion matrix results of the GujiBERT-GCN-LSTM model on the test set. From the confusion matrix, it can be seen that the GujiBERT-GCN-LSTM model performs better on history texts, with 727 samples correctly classified as history and 100 history texts incorrectly classified as non-history. For non-historical texts, a total of 705 samples were correctly classified, and 68 samples were incorrectly classified. The confusion matrix shows that the GujiBERT-GCN-LSTM model performs well in classifying the history class texts, but there are still some misclassifications in the non-history class texts.

To verify the effectiveness of the GujiBERT-GCN-LSTM model proposed in this paper, BERT [19], TextCNN [20], TextRNN [21], FastText [22], DPCNN [23], BERT-CNN [24], BERT-RNN [25], BERT-GCN-LSTM, BERT-BiGRU-CNN [26], TextMGNN [27], SA-SGRU [28], and VGCN [29] were selected for the comparison experiments in this paper.

(1) BERT: Based on a bidirectional Transformer structure, it is capable of capturing contextual information.
(2) TextCNN: Based on a Convolutional Neural Network (CNN), local features of the text are extracted through convolutional layers.
(3) TextRNN: Based on a Recurrent Neural Network (RNN), it captures sequential and dependency information in the text.
(4) FastText: It is a lightweight model that uses averaged word embeddings as input for a simple fully connected layer.
(5) DPCNN: It is a Deep Pyramid Convolutional Neural Network (DPCNN) that extracts deep features layer by layer, suitable for long texts.

(6) BERT-CNN: It combines BERT and CNN structures, utilizing BERT's contextual features and CNN's local feature extraction capabilities.
(7) BERT-RNN: It combines BERT and RNN structures, leveraging BERT's contextual features and RNN's sequential processing capabilities.

Models (1)–(7) are all realized by project https://github.com/649453932/Chinese-Text-Classification-Pytorch (accessed on 10 November 2024).

(8) BERT-GCN-LSTM: This model replaces GujiBERT with BERT based on the GujiBERT-GCN-LSTM structure proposed in this study. It combines BERT, GCN, and LSTM to extract graph structural and sequential features.
(9) BERT-BiGRU-CNN: Proposed by [26], this model integrates BERT's sentence-level feature representations, Bidirectional GRU (BiGRU) for global sequential features, and CNN for local feature extraction, forming an end-to-end text classification model.
(10) TextMGNN: Proposed by [27], this model enhances text classification by introducing multi-granularity topic nodes into the text graph.
(11) SA-SGRU: Proposed by [28], this model combines an improved self-attention mechanism with a Skip-GRU (SGRU) structure.
(12) VGCN: Proposed by [29], this model introduces a variational structure into GCN to learn latent representations of text data.

The ablation experiments are shown in Table 5, and the results of the comparison experiments of the classification model are shown in Table 6.

Table 5. Ablation experiments.

Model	Accuracy	Recall	F1
GujiBERT	0.845	0.940	0.890
GujiBERT-GCN	0.876	0.868	0.872
GujiBERT-GCN-LSTM	0.895	0.914	0.896

Table 6. The comparison of the results of the classification model.

Model	Accuracy	Recall	F1
BERT	0.867	0.859	0.863
TextCNN	0.839	0.827	0.833
TextRNN	0.831	0.793	0.811
FastText	0.854	0.780	0.815
DPCNN	0.816	0.835	0.825
BERT-CNN	0.867	0.864	0.866
BERT-RNN	0.768	0.770	0.769
BERT-GCN-LSTM	0.873	0.848	0.860
BERT-BiGRU-CNN	0.864	0.902	0.870
TextMGNN	0.830	0.861	0.837
SA-SGRU	0.857	0.869	0.860
VGCN	0.824	0.834	0.828
GujiBERT-GCN-LSTM	0.895	0.914	0.896

From Table 5, we can see that GujiBERT has the highest recall and is suitable for application scenarios that need to identify as many samples of the positive class as possible. Still, its accuracy is lower, and there are more misclassifications. GujiBERT-GCN achieves a better balance between accuracy and recall and is suitable for scenarios that require a balance between misclassifications and missed detections. GujiBERT-GCN-LSTM has the highest accuracy, and the primary goal of this topic is ancient text classification to reduce misclassification. Gujibert-GCN-LSTM is the optimal choice, and its F1 value also indicates its better overall performance.

As shown in Table 6, the GujiBERT-GCN-LSTM model achieves the highest performance across all metrics in classifying ancient literature types. This model's superior results suggest that integrating BERT, GCN, and LSTM layers effectively captures complex semantic relationships within ancient texts, outperforming other architectures such as BERT, TextCNN, and TextRNN.

BERT's powerful language representation makes it reliable for general classification. However, it lacks the additional structural and sequential layers that provide GujiBERT-GCN-LSTM with an advantage in capturing the complex linguistic features of the ancient literature. TextCNN and TextRNN classify moderately well, reflecting CNN's proficiency in local feature extraction and RNN sequence processing. Nonetheless, these models lack the pre-trained language understanding and deeper contextualization provided by BERT, which is crucial for nuanced literary genres. FastText has relatively high accuracy but low recall, suggesting a high degree of accuracy but an inability to capture all relevant instances. The DPCNN operates with balanced metrics with the help of its deeper network structure but still lacks the pre-training context provided by the structural insights of BERT and GCN. Hybrid models integrating BERT with other layers show enhanced performance. BERT-CNN maintains competitive performance close to the BERT baseline, while BERT-BiGRU-CNN achieves high recall due to BiGRU's ability to capture bidirectional context and CNN feature extraction. The BERT-RNN model performs poorly, and the RNN layer may not be sufficient to utilize BERT's context embedding effectively. TextMGNN and A-SGRU show competitive results. TextMGNN achieves high recall but low accuracy, which means it performs well in recognizing relevant instances but with low precision. SA-SGRU has a good balance of performance but lacks the fine-grained context sensitivity of the GujiBERT-GCN-LSTM.

4.4. Experimental Results of the Translation Model

4.4.1. Concat Options

There are several different strategies for splicing Gujibert and Entropy-SkipGram word vectors, and to explore which splicing method is more effective, the following standard splicing methods were chosen for comparative experiments: simple concatenation, weighted sum, self-attention fusion, gated mechanism fusion, nonlinear combination, and convolutional layer function. We uniformly set beam size to be 5. The experimental results are measured by the BLEU value.

The results in Tables 7 and 8 provide an insightful comparison of the different fusion methods used to combine Entropy-SkipGram and GujiBERT embeddings, which are analyzed by the BLEU scores for the historical and non-historical datasets. For both types of data, simple concatenation consistently provides the highest BLEU scores, 29.5 for the historical data and 21.0 for the non-historical data, suggesting that directly connecting the two embeddings without any complex transformations tends to retain the most helpful information, thus improving translation performance.

Table 7. Historical concat options.

Method	BLEU
Simple Concatenation	29.5
Weighted Sum	29.3
Self-Attention Fusion	29.4
Gated Mechanism Fusion	29.2
Nonlinear Combination	29.3
Convolutional Layer Function	29.1

In contrast, the more complex fusion techniques, weighted sum, self-attention fusion, and gated mechanism fusion, show slightly lower BLEU scores, although they allow for more dynamic interaction between embeddings. For historical data, these methods hovered between 29.2 and 29.4; non-historical data ranged between 20.7 and 20.9. This suggests that while these methods introduce more flexibility by adjusting the importance of different features, they may not be as effective as simple joins for this particular task. Nonlinear combination and convolutional layer fusion produced the lowest BLEU scores, especially in the non-historical dataset, where they reached 20.7 and 20.8, respectively. This suggests that adding additional layers of complexity through nonlinear transformations or convolutional operations may dilute relevant information in the embedding rather than enhance it. Therefore, this paper chooses the simple connection method to connect Entropy-SkipGram and GujiBERT vectors.

Table 8. Non-historical concat options.

Method	BLEU
Simple Concatenation	21.0
Weighted Sum	20.8
Self-Attention Fusion	20.9
Gated Mechanism Fusion	20.9
Nonlinear Combination	20.7
Convolutional Layer Function	20.8

4.4.2. Beam Size Selection

By setting different beam sizes for the machine translation experiments of the Entropy-SkipGram, the beam size with the best translation effect was selected for the subsequent experiments to achieve better translation performance. The beam sizes set for this experiment are 1, 2, 3, 4, and 5. Table 9 shows the beam size comparison experiments for the historical dataset, and Table 10 shows the beam size comparison experiments for the non-historical dataset.

Table 9. Historical beam size comparison.

Beam Size	BLEU
1	29.1
2	29.4
3	29.3
4	29.3
5	29.5

Table 10. Non-historical beam size comparison.

Beam Size	BLEU
1	20.9
2	21.0
3	21.1
4	21.2
5	21.0

As seen from Table 9, in the translation of historical texts, moderately increasing the beam size will improve the BLEU value, and the BLEU value reaches the highest value of 29.5 when the beam size is 5. Through Table 10, in the translation of non-historical texts, the increase in the beam size has a relatively limited enhancement on the BLEU value, and the BLEU value slightly decreases when the beam size is 5. This indicates that a beam size that is too large is not conducive to improving the translation quality of non-historical texts. Therefore, the beam size of 5 was chosen for historical text translation, and the beam size of 4 was selected for non-historical text translation.

4.4.3. Contrast and Ablation Experiments

Translation model evaluation metrics were adopted from BLEU, and Entropy-SkipBERT selected comparison models were chosen as SkipGram, CBOW, RNNLM, and the OpenNMT default word vector training model.

An analysis of the results presented in Figures 5 and 6 indicates that the Entropy-SkipBERT demonstrates superior efficacy in word vector training for ancient texts, mainly showing significant improvement in elevated training steps. The improvement in the historical translation model is notably more critical than that of the non-historical category. This is because the linguistic structures in historical texts are more uniform, allowing the model to capture word dependencies and contextual relationships more effectively. Furthermore, the historical translation model converges faster and achieves a higher BLEU score than the non-historical model, which can be attributed to data quality issues. Specifically, the average length difference between bilingual pairs is 15 in non-historical documents and also 15 in historical sentences. The average bilingual sentence pair length difference for non-historical documents is 15, and the average bilingual sentence pair length difference for historical documents is 9. The large word count difference implies that there is a large difference in the length of the source- and target-language sentences, which can make sentence alignment difficult. The model needs more time and resources to process these asymmetric sentence pairs, which affects the efficiency and speed of the training. Significant differences in word counts may cause problems with the syntax and fluency of generated translated sentences and increase the inconsistency of lexical alignment. This inconsistency affects the degree of matching of individual n-grams in the BLEU score, decreasing the BLEU score.

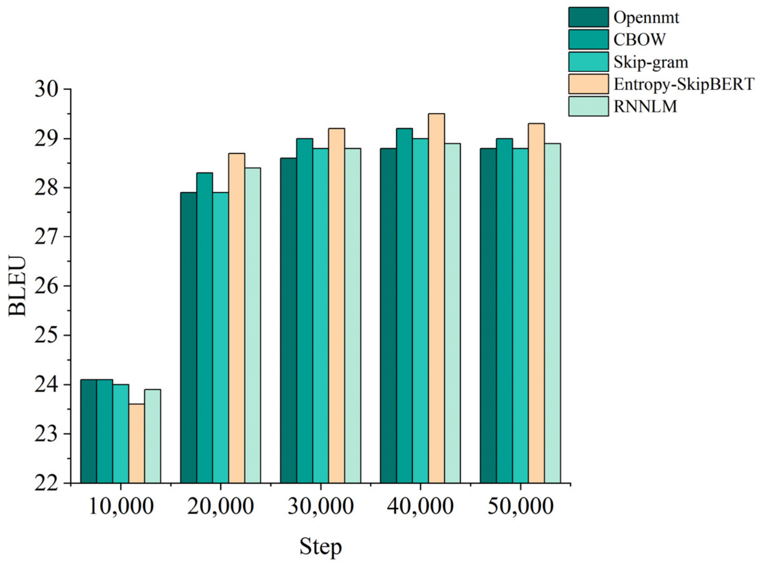

Figure 5. Performance comparison of history class translation models.

To validate the effectiveness of this model, experiments were conducted on the following machine translation models using the same dataset.

(1) Transformer [30]: Proposed by Vaswani et al., this model is based entirely on a self-attention mechanism. With multi-head self-attention and positional encoding, it processes sequences in parallel, capturing long-range dependencies efficiently and with high performance.

(2) Transformer–PASCAL [31]: Proposed by Bugliarello et al., this model introduces a parameter-free Parent-Scaled Self-Attention (PASCAL) module into the Transformer's self-attention mechanism, allowing the model to focus on the dependency parent node of each word, thus incorporating syntactic information when encoding the source sentence.

(3) DTCL [13]: Proposed by Zheng et al., this model combines dependency syntax trees with LSTM, introducing syntactic structure information of the source language. Generating dependency relationship matrices with dependency syntax trees and using LSTM for positional encoding enhance the model's ability to capture sequential information.

(4) GCN–Attention Machine Translation Model [32]: Proposed by Chai et al., this model combines attention mechanisms within a GCN to enhance semantic understanding.

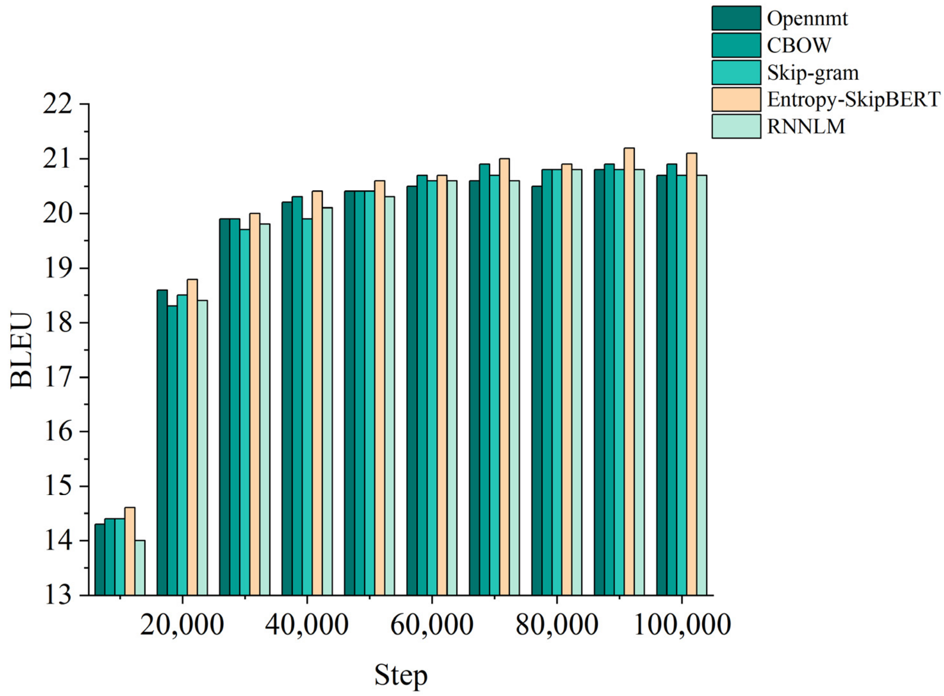

Figure 6. Performance comparison of non-history class translation models.

As shown by the BLEU scores in Table 11, the two-stage ancient text translation model proposed in this paper achieved the highest score, indicating that its word matching and phrase selection are closer to the reference translation, making it well suited for the complexity of ancient text translation. In comparison, the base Transformer model has a lower BLEU score due to its lack of deep syntactic structure modeling. Transformer + PASCAL, with the addition of the PASCAL module and DTCL, which incorporates dependency syntax trees, shows some improvements in capturing dependency relations but with limited effect. The GCN–Attention model performs well in handling local relationships, with a slight increase in the BLEU score. Overall, the two-stage model better captures ancient texts' unique dependencies and complex semantics.

Table 11. Comparison experiment of machine translation models.

Model	BLEU
Transformer	20.1
Transformer + PASCAL	20.8
DTCL	20.9
GCN–Attention Machine Translation Model	21.2
Two-Stage Ancient Text Translation Model	22.8

The experimental results of the translation model are shown in Table 12, where T denotes the Transformer model, E denotes the Entropy-SkipBERT model introduced, and G denotes the inclusion of the classification model.

As shown in Table 12, the two-stage translation model achieves a 2.7-BLEU-point improvement over the benchmark model. The translation model proposed in this paper demonstrates its effectiveness in enhancing the translation of ancient texts. Furthermore, incorporating the dependency matrix into word vector training for both types of literature translation models results in increases of 0.7 and 0.4 BLEU points, respectively, indicating that the dependency matrix contributes to improved translation performance. The translation performance of the historical literature translation model is higher than that of the non-historical literature. When conducting a thorough analysis of the dataset, it is evident that the vocabulary and grammar of the historical literature exhibit a notable degree of stability, which is conducive to learning the model.

Table 12. Ablation experiment.

Model	BLEU
T-mix	20.1
T + G-mix	22.2
T + E + G-mix	22.8
T-histories	28.8
T + E-histories	29.5
T-non-historical	20.8
T + E-non-historical	21.2

Compared to historical texts, the model faces distinct challenges in translating non-historical texts. Non-historical texts exhibit greater lexical diversity, including modern terminology, colloquial language, and specialized jargon, complicating accurate decoding and semantic interpretation. Additionally, these texts often use informal syntactic structures—such as ellipses, inversions, and regional phrases—that challenge the model's parsing capabilities. Non-historical texts also feature nuanced emotional and tonal shifts, which complicate sentiment recognition, and their complex contextual structures frequently include metaphors, subjective evaluations, and abrupt shifts, all of which require inferential solid reasoning. Furthermore, ambiguous references are more common, making referent resolution more difficult. These factors reduce the model's accuracy and fluency in translating non-historical compared to historical texts.

4.4.4. Example Analyses

This paper introduces an example analysis, which is carried out on historical texts and non-historical texts, respectively, to evaluate the performance of the translation model more comprehensively. For comparison, the translation results of the selected word embedding models are a random vector, CBOW, RNNLM, and Entropy-SkipBERT. Table 13 shows the results of analyzing the historical examples, and Table 14 shows the results of analyzing the non-historical examples.

Table 13. Example analysis of history class.

Model	Translation Results	Translation Results in English
Original text	会参掌虽繁，教授不阙。	Though the palm of the Council is very complicated, the professor does not have to do anything about it.
Original reference	虽然权会掌管的事情很多，但从不中断教书。	Although there are many responsibilities to manage, teaching is never interrupted.
Random vector	恰巧参掌管虽然，教授不缺。	By coincidence, despite participation in management, there is no shortage of teaching.
CBOW	恰逢参与掌管制度，教授没有缺。	Just in time to participate in the management system, there is no shortage of teaching.
RNNLM	碰上参与掌繁琐，教授门不阙。	Encountering involvement in complex tasks, the teaching does not fall short.
Entropy-SkipBERT	适逢参与掌管繁琐的事务，教授不缺。	At the right time to participate in managing intricate tasks, teaching is not lacking.

Table 14. Example analysis of non-history class.

Model	Translation Results	Translation Results in English
Original text	圣君独有之，故能述仁义于天下。	It is unique to the saintly ruler, so he is able to describe benevolence and righteousness to the world.
Original reference	只有圣明的君主明白这个道理，所以他在天下能继承仁义。	Only an enlightened ruler understands this truth, so he can uphold righteousness in the world.
Random vector	圣明的君主只是有这<unk>，所以能在于天下自称道仁义。	The enlightened ruler just has this <unk>, so he can claim righteousness in the world.
CBOW	圣明君只有这种事，所以能够在天下的道义中。	The wise ruler has only this matter, so he can be in the righteousness of the world.
RNNLM	圣贤的君主独自有它，所以能够对天下的道理。	The sage ruler alone has it, so he can follow the principles of the world.
Entropy-SkipBERT	圣贤的君主独自有此事，所以能够述周天下的仁义道。	The wise ruler alone has this matter, so he can spread the path of righteousness around the world.

From the analysis of historical examples in Table 13, it can be seen that the original text of ancient Chinese expressed the concept of uninterrupted teaching even with numerous management responsibilities. The reference translation emphasizes the importance of teaching despite the heavy workload. The translation of the random vector model incorporates the unnecessary element of “恰巧 (coincidental)”, causing the translation to lose coherence and fail to express the core meaning that has never been ignored in teaching. The translation of the CBOW model mentions “制度 (system)”, which is not mentioned in the original text and fails to reflect the balance between teaching and management responsibilities accurately. Although the RNNLM model retains the concept of “繁琐 (tedious)” translation, its fluency is poor, and it does not fully convey the meaning of always adhering to teaching. RNNLM translation is relatively smooth, but the translation of words in sentences is not precise enough, resulting in a semantic deviation from the original text. The translation of the Entropy-SkipBERT model accurately captures the meaning of “参与管理繁琐事务 (participating in managing tedious affairs).” It expresses the meaning of teaching not being ignored, which is closer to the original text. Entropy-SkipBERT performs the best in accuracy and fluency and is closest to reference translation.

In Table 14, the original text emphasizes that only wise monarchs possess the unique quality of spreading benevolence and righteousness, reflecting the importance of moral leadership. The reference translation effectively conveys the monarch’s wisdom and the ability to spread virtues. The <unk> symbol in the translation of the random vector model damages the integrity of the sentence, and “自称道仁义 (claiming to be righteous and benevolent)” carries a boastful connotation, misunderstanding the main purpose of the monarch’s dissemination of benevolence and righteousness. The translation of the CBOW model is relatively vague and fails to accurately reflect the unique quality of monarchs actively spreading benevolence and righteousness. The translation of the RNNLM model introduces the concept of “道理 (reason)”, deviating from the core meaning of “传播仁义 (spreading benevolence and righteousness)”, and only expressing the static meaning of the ruler having certain characteristics. In contrast, the Entropy-SkipBERT model accurately reproduces the original text and represents the theme of monarchs spreading benevolence and righteousness to the world through their unique qualities. It outperforms other models in both accuracy and fluency.

The analysis of these two tables shows that the Entropy-SkipBERT model performs the best in capturing semantics and maintaining sentence fluency. In contrast, other models often deviate from the original meaning, introduce redundant elements, or express themselves unclearly.

5. Conclusions

This paper explores classification and machine translation methods for Chinese ancient texts. First, the framework adopts the GujiBERT-GCN-LSTM model to classify and predict ancient texts automatically, dividing them into historical and non-historical datasets based on classification results and training separate translation models for each type. Using a neural network translation architecture, we introduced an entropy-weighted Skip-Gram model, combined with the GujiBERT word vector model, to optimize the translation model's training process. Experimental results show that the GujiBERT-GCN-LSTM model achieves high classification accuracy. Meanwhile, the Entropy-SkipBERT model demonstrates superior machine translation across both datasets by accurately capturing semantic information, enhancing syntactic understanding, reducing noise word influence, and adapting to text styles, ultimately contributing to improved translation quality. Overall, our proposed two-stage model significantly optimizes translation results compared to traditional models.

This model can automatically translate ancient texts into modern Chinese by digitizing ancient texts and building digital archives, greatly increasing translation efficiency and reducing human effort. This process aids in preserving and disseminating the traditional literature, providing accessible, modernized content for ancient text enthusiasts and educators. Considering the complex chapter structure and rich contextual associations of ancient texts, future research can integrate this contextual information to further improve the translation model's architecture and overall performance in ancient text translation.

Author Contributions: The authors confirm contribution to the paper as follows. Study conception and design: F.Y., R.H., Y.Z. and Y.H.; data collection: R.H.; analysis and interpretation of results: F.Y. and R.H.; draft manuscript preparation: R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Reform Project of Machine Translation Scientific Application of English for Cybersecurity Master's Degree at North China University of Science and Technology, grant number 221006043084307.

Data Availability Statement: The data that support the findings of this study are openly available in Classical-Modern at https://github.com/NiuTrans/Classical-Modern and Chinese Poetry at https://github.com/chinese-poetry/chinese-poetry accessed on 9 November 2024.

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

References

1. Hu, S.; Li, X.; Bai, J.; Lei, H.; Qian, W.; Hu, S.; Zhang, C.; Kofi, A.S.; Qiu, Q.; Zhou, Y. Neural Machine Translation by Fusing Key Information of Text. *Comput. Mater. Contin.* **2023**, *74*, 2803–2815. [CrossRef]
2. Wang, D.; Liu, C.; Zhao, Z.; Shen, S.; Liu, L.; Li, B.; Hu, H.; Wu, M.; Lin, L.; Zhao, X.; et al. Gujibert and Gujigpt: Construction of Intelligent Information Processing Foundation Language Models for Ancient Texts. *arXiv* **2023**, arXiv:2307.05354.
3. Wang, K.; Han, S.C.; Poon, J. Induct-GCN: Inductive Graph Convolutional Networks for Text Classification. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 1243–1249.
4. Jang, B.; Kim, M.; Harerimana, G.; Kang, S.; Kim, J.W. Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism. *Appl. Sci.* **2020**, *10*, 5841. [CrossRef]
5. Malik, S.; Sushma, N.S.; Sharma, S.K. Effect of GloVe, Word2Vec and fastText Embedding on English and Hindi Neural Machine Translation Systems. In Proceedings of the Data Analytics and Management: ICDAM 2022, Singapore, 21–23 September 2022; pp. 433–447.
6. Krimpas, P.; Valavani, C. Attention Mechanism and Skip-Gram Embedded Phrases: Short and Long-Distance Dependency N-Grams for Legal Corpora. *Comp. Legilinguist.* **2022**, *52*, 318–350. [CrossRef]
7. Riess, S.; Huck, M.; Fraser, A. A Comparison of Sentence-Weighting Techniques for NMT. In Proceedings of the Machine Translation Summit XVIII: Research Track, Online, 16–20 August 2021; pp. 176–187.
8. Li, Z. Research on Aspect-Based Sentiment Analysis Technology Based on Multi-Task Learning. Master's Thesis, Qilu University of Technology, Jinan, China, 2024.
9. Wu, M.; Lin, L.; Hu, D.; Liu, C.; Huang, S.; Meng, K.; Wang, D. Research on Machine Translation from the Perspective of Temporal Characteristics in Ancient Chinese Classical Texts. *Libr. Trib.* **2024**, *44*, 93–102.
10. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Veselin, S. Roberta: A Robustly Optimized Bert Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
11. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv* **2019**, arXiv:1910.01108.
12. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; Hu, G. Pre-Training with Whole Word Masking for Chinese BERT. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3504–3514. [CrossRef]

13. Xin, Z.; HaiLong, C.; Yuqun, M.; Qing, W. Neural Machine Translation Model Combining Dependency Syntax and LSTM. *J. Harbin Univ. Sci. Technol.* **2023**, *28*, 20–27.
14. Xi, N.; Lijia, Z.; Lutong, W.; Jun, C.; Xiaorong, W. A Method for Implementing Word2Vec to Construct Word Embedding Model. *Comput. Inf. Technol.* **2023**, *31*, 43–46.
15. Feng, Y.; Hu, C.; Kamigaito, H.; Takamura, H.; Okumura, M. A Simple and Effective Usage of Word Clusters for CBOW Model. *J. Nat. Lang. Process.* **2022**, *29*, 785–806. [CrossRef]
16. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–9.
17. Cognetta, M.; Hiraoka, T.; Okazaki, N.; Sennrich, R.; Pinter, Y. An Analysis of BPE Vocabulary Trimming in Neural Machine Translation. *arXiv* **2024**, arXiv:2404.00397.
18. Klein, G.; Hernandez, F.; Nguyen, V.; Senellart, J. The OpenNMT Neural Machine Translation Toolkit: 2020 Edition. In Proceedings of the 14th Conference of the Association for Machine Translation in the Americas, Online, 6–9 October 2020; Volume 1: Research Track, pp. 102–109.
19. Abarna, S.; Sheeba, J.I.; Devaneyan, S.P. An Ensemble Model for Idioms and Literal Text Classification Using Knowledge-Enabled BERT in Deep Learning. *Meas. Sens.* **2022**, *24*, 100434. [CrossRef]
20. Zhang, T.; You, F. Research on Short Text Classification Based on TextCNN. *J. Phys. Conf. Ser.* **2021**, *1757*, 012092. [CrossRef]
21. Liu, Z. Text Classification of Electricity Policy Information Based on BERT-Optimized TextRNN. In Proceedings of the 2022 3rd International Conference on Computer Science and Management Technology (ICCSMT), Shanghai, China, 18–20 November 2022; pp. 76–79.
22. Mojumder, P.; Hasan, M.; Hossain, M.F.; Hasan, K.A. A study of fasttext word embedding effects in document classification in bangla language. In *Cyber Security and Computer Science: Proceedings of the Second EAI International Conference, ICONCS 2020, Dhaka, Bangladesh, 15–16 February 2020*; Proceedings 2; Springer International Publishing: Cham, Switzerland, 2020; pp. 441–453.
23. Yu, S.; Liu, D.; Zhang, Y.; Zhao, S.; Wang, W. DPTCN: A novel deep CNN model for short text classification. *J. Intell. Fuzzy Syst.* **2021**, *41*, 7093–7100. [CrossRef]
24. Wan, C.X.; Li, B. Financial causal sentence recognition based on BERT-CNN text classification. *J. Supercomput.* **2022**, *78*, 6503–6527. [CrossRef]
25. Shreyashree, S.; Sunagar, P.; Rajarajeswari, S.; Kanavalli, A. BERT-based hybrid RNN model for multi-class text classification to study the effect of pre-trained word embeddings. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 667–674.
26. Rong, Z.; Gang, Q. Research on the Classification of Satirical Texts Based on Bert-Bigru-Cnn. *Intell. Comput. Appl.* **2024**, *1*, 1–6.
27. Gu, Y.; Wang, Y.; Zhang, H.-R.; Wu, J.; Gu, X. Enhancing Text Classification by Graph Neural Networks with Multi-Granular Topic-Aware Graph. *IEEE Access* **2023**, *11*, 20169–20183. [CrossRef]
28. Huang, Y.; Dai, X.; Yu, J.; Huang, Z. Sa-SGRU: Combining Improved Self-Attention and Skip-GRU for Text Classification. *Appl. Sci.* **2023**, *13*, 1296. [CrossRef]
29. Ren, Z. VGCN: An Enhanced Graph Convolutional Network Model for Text Classification. *J. Ind. Eng. Appl. Sci.* **2024**, *2*, 110–115.
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
31. Bugliarello, E.; Okazaki, N. Enhancing Machine Translation with Dependency-Aware Self-Attention. *arXiv* **2019**, arXiv:1909.03149.
32. Hua, C. Research on Machine Translation Method Based on Graph Convolution Neural Network and Attention Mechanism. *Autom. Instrum.* **2023**, *10*, 187–190.