WILEY | Hindawi

Retraction

Retracted: Analysis and Application of the Business English Translation Query and Decision Model with Big Data Corpus

Security and Communication Networks

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Gong, "Analysis and Application of the Business English Translation Query and Decision Model with Big Data Corpus," *Security and Communication Networks*, vol. 2022, Article ID 2714079, 10 pages, 2022.

WILEY | Hindawi

Research Article

Analysis and Application of the Business English Translation Query and Decision Model with Big Data Corpus

Jufang Gong (ID)

School of Foreign Studies, Xiangsihu College of Guangxi Minzu University, Nanning 530225, China

Correspondence should be addressed to Jufang Gong; army2009@gxmzu.edu.cn

This paper aims to build an English translation query and decision support model using big data corpus and applies it to business English translation. Firstly, the existing convolutional network is improved by using depth-separable convolution, and the input statements are mapped to the depth feature space. Secondly, the attentional mechanism is used to enhance the expressive ability of input sentences in deep feature space. Then, considering the sequential relationship, use long short-term memory (LSTM) neural network as a decoder block to generate the corresponding translation of the input sentence. Finally, nonparametric metric learning module is used to improve the model in an end-to-end way. Wide range of experiments on the multiple corpora have shown the proposed model has better real-time performance while maintaining high precision in translation and query, and it has a certain practical application value.

1. Introduction

The new technologies are developing rapidly, such as big data and artificial intelligence (AI), machine translation is becoming more intelligent and humanized, and the results of machine translation are gradually approaching the human mind manner. The error rate of machine translation has gradually decreased, and the accuracy of translation has made a breakthrough [1] by using big data, AI, natural language processing, and other related technologies. Microsoft, Baidu, and Google have launched AI-based intelligent translation systems. However, the machine translation based on AI technology has made great progress, and there are still many shortcomings in terms of deep semantic structure, different stylistic styles, language styles, and discourse levels, which are hard to satisfy the needs of practical application [2]. Therefore, we need to build a corpus with big data. And the constructed corpus is used to train the translation model, which makes machines increasingly intelligent.

The corpus-based method is to describe the same information or topic in two or more languages and establish the connection between different language information through manual or computer. Corpus is divided into parallel corpus and comparable corpus [3]. Corpus has mostly obtained information resources through the Internet and large literature collection and analyzed and constructed automatically. The corresponding relationship between different languages on the same information or topic in the corpus is aligned, which contains abundant translation materials and knowledge. For parallel corpus, the query translation process will produce different results, through the statistical probability of the word in both source and target languages in word frequency. It can select the best translation results by using statistical analysis. The main methods include cross-language latent semantic indexing, generalized vector space model, and correlation feedback method. The above method aims to achieve fast indexing and eliminate ambiguity between words. Different from dictionaries, this kind of method has a large amount of semantic space. Corpus exists in chapters, abstracts, and paragraphs, while parallel corpus has a powerful computing function, providing a good platform for translation queries in cross-language information retrieval [4].

Query translation is the language used to directly translate users' query requirements into target documents.

This approach is currently the most widely used translation strategy and can be realized by using a single language retrieval system directly [5]. The advantage of this approach is simplicity and flexibility, and it only needs a small resource consumption, but there will be polysemy in the process of user query translation. To solve the ambiguity problem, the statement package is added first, and the query expression is extended. In addition, the system interaction function is used to achieve the query requirements. Secondly, the recognition and matching of the input statements are completed with the theme meaning of the file, and the natural language processing mode is adopted. Finally, the feedback results are re-translated. The following techniques are generally used to implement questioning translation: dictionary-based approach, corpus-based approach, dictionary-corpus hybrid approach, and questioning weighting method.

The core modules of the translation and decision system mainly include corpus construction and decision module [6]. Among them, the corpus is the carrier of knowledge representation and storage. At present, the corpus mainly expresses the semantic relationship between the sentence to be queried and empirical knowledge by using triples [7]. The query and decision-making task of the corpus is to identify the natural language which included entities, entity relationships, and entity types. The answers are searched based on the language matching manner [8]. Recently, common translation query and decision-making systems mainly include semantic parsing methods and information retrieval. The semantic parsing method directly identifies entities, entity relationships, and entity combinations from questions by compiling rule bases, auxiliary dictionaries, artificial reasoning, machine learning, and deep learning. For example, Cheng et al. [9] have used the sequence labeling model to identify the entities in the problem, used the sequence-to-sequence model to predict the relational sequence in the problem, and used the answer verification mechanism and the circular training method to improve the performance of the model. Moradshahi et al. [10] have proposed an automatic machine translation generator based on large-scale dialogue data sets to solve the problem that the multi-task dialogue system is not robust. The input sentences are analyzed according to the similarity value between the corpus and the sentences.

The method based on information retrieval firstly identifies the candidate entity set in the question by other means, such as entity identification technology and entity dictionary, and then, according to the predefined logical form, all the one-hop or multi-hop relationships of candidate entities in the knowledge base are queried from the corpus, thus obtaining the candidate query path set. Finally, the candidate query path with the highest matching degree is obtained by calculating the similarity value among the query path and the problem, and the answer is queried in the knowledge base [11]. For example, Zhang et al. [12] have presented a method to enhance relationship matching, bi-layer bidirectional long short-term memory network (BiLSTM) is first used to match candidate relationships at multiple levels, and then it uses relationship matching to reorder entity link results. Li et al. [13] have simplified the matching mode of the existing question answering system by using a knowledge map, by convolutional neural network for identifying the semantic features of questions, by the matching degree of answers and questions for determining the results. Li et al. [14] proposed a causal relationship extraction model for a knowledge convolution neural network. The causal relationship in the input sentence has captured by fusing human prior knowledge, and the attribute mapping has performed based on the BiLSTM of the attention mechanism. Finally, the answer was selected from the knowledge base based on the results of the first two steps.

The number and variety of information resources available on the Internet are becoming increasingly rich, and the language is uneven and diverse. The number of Internet users is also growing, and their languages are diverse. Due to the diversity of language resources on the Web and the differences in language mastery among Web users, language barriers arise when retrieving information via the Web, causing inconvenience to non-English speaking users. Thus, this has important implications for the study of English-Chinese cross-language information retrieval design.

In conclusion, although all the above models have achieved good translation results, most of them are only verified on short sentences, and their effects on long sentences and complex sentences are not good. Secondly, the overall performance of business, medical, and other professional translation is poor. The main reason is that the contextual semantic information is not fully considered. In addition, professional databases such as medical and business English contain limited words. Therefore, this paper will build a corpus-based English translation query and decision system, which is based on big data, and apply it to business English translation.

2. Model Design

Figure 1 shows the analysis and application of the business English translation query and decision system supported by the big data corpus. (a) A feature extraction module extracts the depth features of input text at character and word level by using a convolutional neural network; (b) attention mechanism considers the context sequence relationship of the input text, using the attention mechanism to strengthen the expressive ability of features in-depth space; (c) a decoding module decodes the output features of the attention mechanism to generate a translation corresponding to the input text; (d) measurement learning module: the cosine similarity between the translation generated by the decoder and the corresponding real tag is measured to realize the end-to-end optimization of the model.

2.1. Textual Vectorization. Text vectorization is the first step of the translation query system. Common text vectorization includes one-hot representation, TF-IDF (term frequency-inverse document frequency) representation, and Word2-Vec representation.
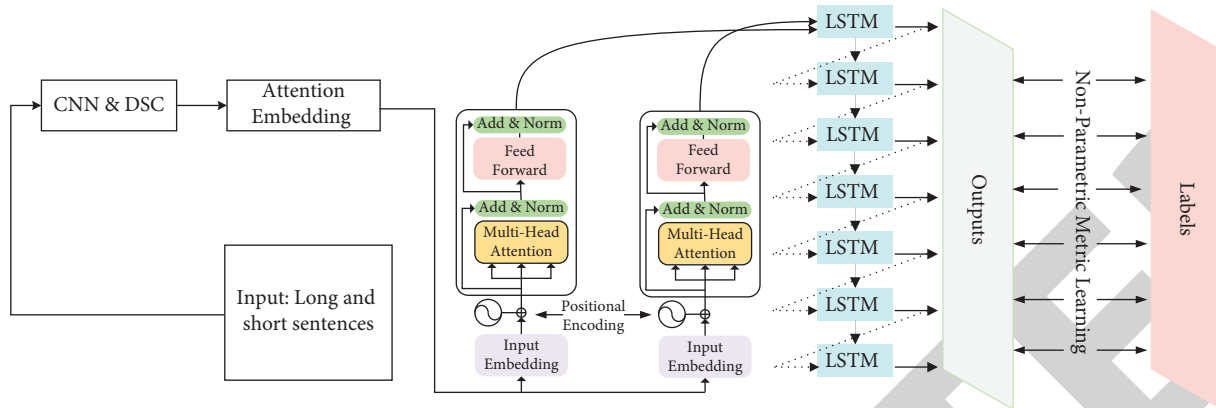
FIGURE 1: Model architecture diagram.

One-hot representation is the earliest and intuitive word vector generation method with a simple and direct structure, and the generated vector representation reflects the information of word frequency. This mapping method firstly summarizes all the words in the corpus to obtain N-words and generates an N-dimensional vector for each document in the corpus, in which each dimension reflects how many specific words exist in the document. However, this textual vectorization only considers word frequency and causes the vector lengths of long sentences and short sentences to be inconsistent.

TF-IDF represents the mapping relationship that takes into account the inverse document frequency in the document vocabulary. When using the TF-IDF method for text vectorization, the word frequency is normalized first, even though the frequency of word occurrence rather than the number represents the word frequency. In addition, the existing improved TF-IDF text vectorization method also considers the inverse document frequency index of each word and uses this index as a measure of word rarity to better characterize the document generation vector. However, the common disadvantage of the above two models is that the vector length is very large. For a corpus with a large vocabulary, the mapping vector length of each document is long, which means that the resulting matrix is very sparse and complicated to calculate.

In natural language processing (NLP), Word2Vec representation is the most classical pertaining models. Word2Vec has the advantages of simplicity, speed, and versatility, but it is limited by the corpus, and the word representation generated is static, unable to solve the problem of polysemy, and cannot reflect the multi-layer characteristics of the word, including grammar and semantics.

BERT model is a new vectorized representation of text processing proposed in recent years. Compared with word embedding methods represented by Word2Vec, the BERT model is more adaptive and can solve the problem of polysemy. In addition, BERT based on transformer, using transformer's bidirectional encoder structure, can reflect the multi-layer characteristics of words. This paper uses the

BERT-based model, which has 12 transformer encoders, each transformer encoder has 12 heads, and the hidden layers have a size of 768.

2.2. Feature Extraction. Convolutional neural network (CNN) is a feed-forward neural network that has been widely used in recent years in text classification, computer vision, and other fields [15]. The translation query and decision system constructed in this paper firstly use a convolution network as a feature extractor to realize the feature extraction of input text. CNN is mainly divided into convolutional layer, pooling layer, and fully connected layer. Here, inspired by bidirectional short memory neural networks, we take full account of the fact that the model will analyse two sentences in parallel manner. Therefore, we use the parallel convolutional neural network structure here. Compared with traditional convolution, it has a better sensibility field due to the consideration of parallel spatial sequences. However, as the depth of the network increases, the time cost of the model will inevitably increase. This paper comprehensively considers the real-time requirements of the translation system. Here, we adopt parallel convolution to alleviate the time cost problem. Thus, the parallel CNN structure is adopted. The framework of the parallel CNN is presented in Figure 2.

In Figure 2, each layer of the network is responsible for realizing different functions, among which the convolution layer is responsible for completing most of the computation work in the convolution network, such as local feature extraction, thus making CNN generate a local receptive field. The pool layer is usually located between two convolution layers. By down-sampling, the number of parameters and network calculations is gradually reduced, and the spatial dimension is reduced. The typical pooling method is to calculate the maximum value of each local block in the feature matrix, and then the adjacent pooling units are translated to get the feature matrix of the lower input. The specific operation is shown in formula.

$$\begin{cases} c_i = \sigma(w\mathbf{X} + b), \\ \mathbf{X} = [x_1, x_2, \ldots, x_{i+1}, x_{i+\kappa}], \end{cases} \quad (1)$$

where c_i is the result of the convolution calculation, X is the input matrix, σ indicates the activation function,
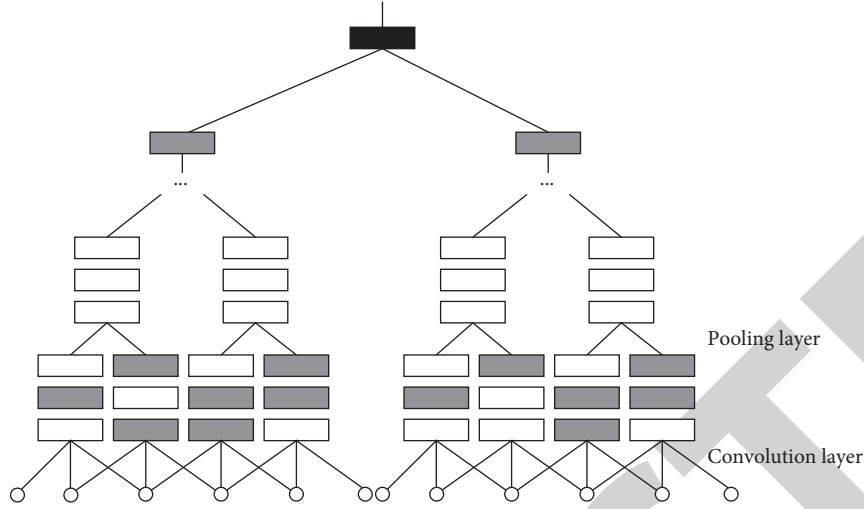
Figure 2: Parallel CNN structure.

PReLU (parametric corrected linear unit) activation function is used here, w represents the parameter of each convolutional kernel, and κ indicates the size of the convolutional kernel.

Here, considering the high real-time requirement of the actual translation and query system depthwise convolution (DW) [16] and pointwise convolution (PC) [17] are combined to form depthwise separable convolution so as to reduce the computational burden of the model and further reduce the time cost of the model by reducing the number of parameters. In particular, depth-separable convolution consists of depthwise convolution (DW) and pointwise convolution (PC). By decomposing the standard convolution process into several equivalent deep convolutions and point-by-point coupon products, the computational complexity of the model is reduced. The processing of the deep-separable convolutional neural network is shown in Figure 3.

In the traditional convolution operation, the convolution kernel of the size $L \times L$ is used in the convolution calculation phase, and the output channel size is M. The deep convolution DW-operated processing is shown in Figure 3. M convolution kernels of the size $L \times L$ are used for each convolution calculation, and the output is usually 1. In every calculation of the pointwise convolution, M convolution kernels of the size are used for convolution filtering. Deep convolution DW and PC can be spliced into a standard convolution with convolution kernel size $L \times L$, and the channel is also set to M. Among them, the parameters involved in the calculation of the standard convolution CNN are shown in formula.

$$\Delta_{\text{CNN}} = L \times L \times M \times N. \tag{2}$$

The number of parameters involved in the depth-separable convolution calculation of the combination of DW and PC is shown in

$$\Delta_{DW,PC} = L \times L \times 1 \times M + 1 \times 1 \times M \times N, \tag{3}$$

$$R = \frac{\Delta_{DW,PC}}{\Delta_{CNN}} = \frac{L \times L \times 1 \times M + 1 \times 1 \times M \times N}{L \times L \times M \times N}$$
$$= \frac{1}{L^2} + \frac{1}{N}. \tag{4}$$

It can be seen from formula (4) that when convolutional kernel $L \geq 2$, the number of parameters involved in the deeply separable convolution calculation is obviously less than that in the standard convolution calculation. Therefore, using the depth-separable convolution instead of the standard convolution can reduce the time cost in convolution calculation.

2.3. Attention Mechanism.
Although depth-separable convolution can extract the deep features of the input text or sentence, it fails to fully consider the contextual semantic information between characters in the input sentence, resulting in poor fluency and incomplete semantic information of the translation results, especially in complex sentences or long sentences. Here, we take inspiration from transformer and use the encoder block and multi-focus module provided by transformer to effectively alleviate the problem of long sentences relying on long distance. The transformer block can be found in Figure 4.

To address the above problems, we introduce transformer encoder block with an attention mechanism, take the output feature map of depth-separable convolution as the input of the transformer coding block, using the attention mechanism provided by the coding block to establish the context dependency [18], and solve the problem that the existing models make insufficient use of deep semantic information. The transformer structure is shown in Figure 4.

In Figure 4, the left part illustrates the encoder block and the right part shows the decoder block. Multiple encoder block is used to form the encoder module. Similarly,
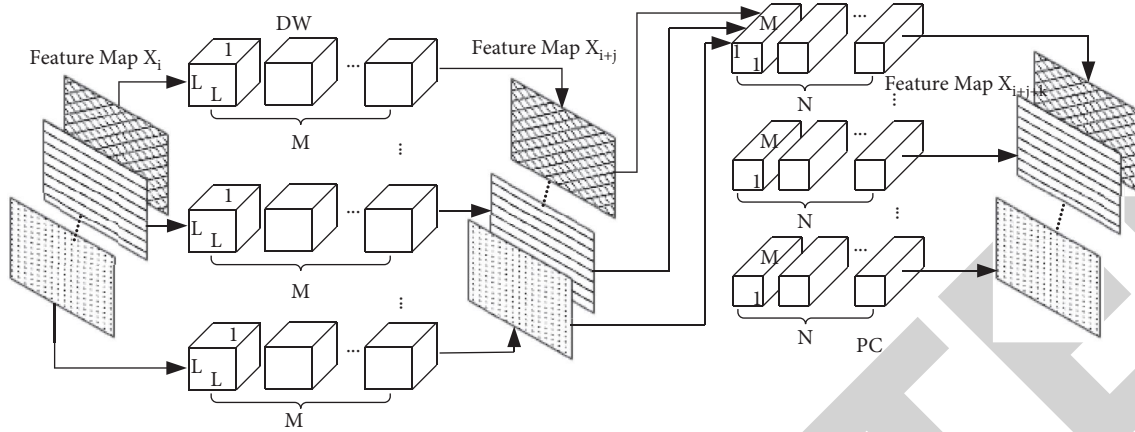
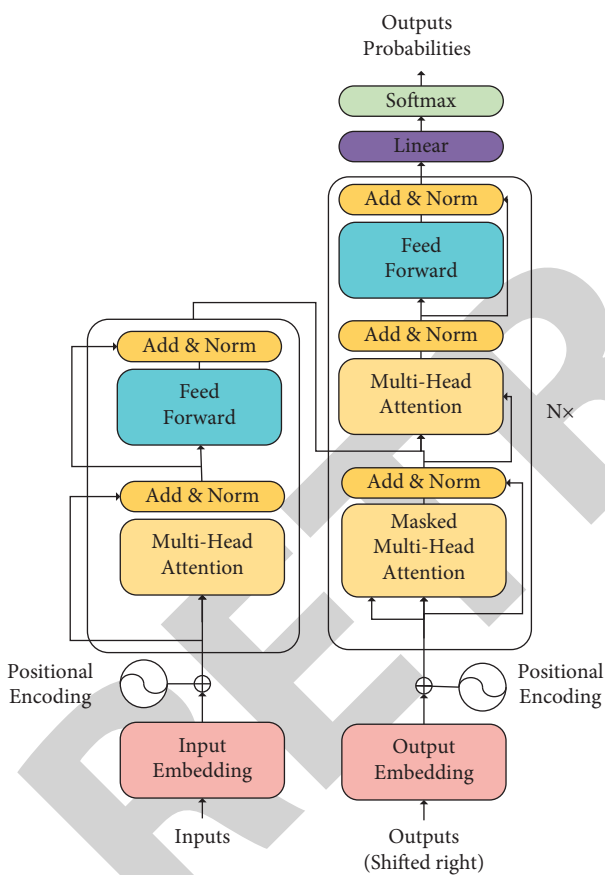FIGURE 3: Structure of depth-separable convolution model.

FIGURE 4: Transformer architecture.

multiple decoder block is used to form the decoder module. The task in Figure 4 will be taken as an example.

The input of the encoder is a natural language. Before the first encoder block is sent, the natural language needs to be digitized. The processing flow is shown in Figure 5. Firstly, word embedding and position embedding of each token are added bit by bit for each token after word segmentation, and the representation vector of each token in the statement is obtained and arranged line by line to form an input matrix X that can represent a sentence. Here, the word embedding of

each token is obtained by the Word2Vec module; the position embedding of each token is shown by formula.

$$
\begin{cases}
PE_{(pos,2i)} = \sin\left(\dfrac{pos}{10000^{2i/d_{model}}}\right), \\
\\
PE_{(pos,2i+1)} = \cos\left(\dfrac{pos}{10000^{2i/d_{model}}}\right),
\end{cases}
\tag{5}
$$

where pos indicates the position of the token in the sentence, $2i$ represents the $2i$-th position in the position vector of the token, and the d_{model} represents the word vector dimension.

2.4. Decoder Module. The text has a strong temporal attribute, and the corresponding words in the translation also have sequence dependence [19]. Considering that the traditional template matching generation method only decodes short sentences, it does not have to deal with long-term dependencies. In addition, we adopt parallel convolution (as shown in Figure 2) in the feature coding stage, and feature decoding after the output of parallel convolution can be realized only by using one-way LSTM network. Since parallel convolution has been used in the encoding stage, i.e., a single convolution is used for feature encoding on each of the two channels, and the semantic information on the adjacent time series has been considered for the upper and lower channels. Therefore, the decoder module of this paper adopts the multi-layer LSTM as the basic structure. LSTM is a special recurrent neural network (RNN), and it is used to solve the gradient disappearance and gradient explosion in long sequence training. Specifically, compared with RNN, LSTM has better performance in long sequence dependence problem. The network structure of the LSTM unit is shown in Figure 6 and consists of an input gate, forgetting gate, output gate, and memory unit [20].

Assume that at time t, the output result of the model encoder is set 1024-dimensional vector x_t. In the previous step, the LSTM gating unit outputs the hidden layer result as h_{t-1}, and the memory unit of the current step is c_t. The activation function is used to weight the feature vectors, and the output result of the input gate is obtained as the input eigenvector of the LSTM unit. Similarly, the forgetting gate
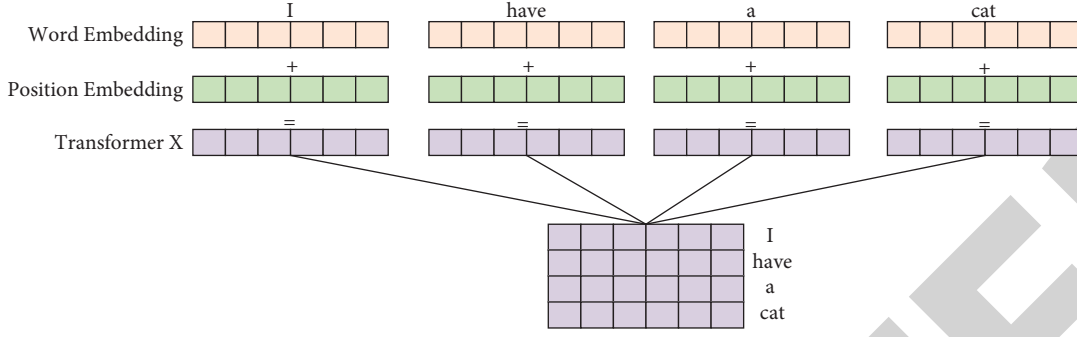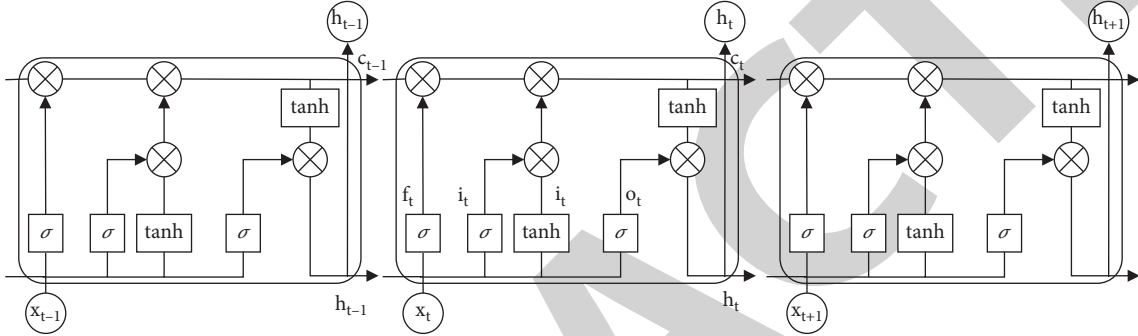
FIGURE 5: Schematic diagram of the word embedding.

FIGURE 6: LSTM network structure.

feature f_t and the output feature o_t can be obtained, and the calculations are shown in

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right), \tag{6}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right), \tag{7}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right), \tag{8}$$

$$g_t = \phi\left(W_{xg}x_t + W_{hg}h_{t-1} + b_g\right), \tag{9}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t, \tag{10}$$

$$h_t = o_t \otimes \phi\left(c_t\right), \tag{11}$$

where σ represents the sigmoid activation function, ϕ is the tanh activation function, and $\otimes$ represents Hadamard product operation.

In the decoding process, the coded feature sequence after the attention mechanism is used as input for the decoder LSTM and then obtains the output result of the corresponding hidden layer $H = (h_1, h_2, h_3, \ldots, h_n)$. When the model decoder inputs the last feature vector, the hidden layer of LSTM network outputs the result, that is, the decoding result of the current input text sequence. LSTM decoder is shown in Algorithm 1.

2.5. *Nonparametric Metric Learning Module.* Considering the real-time requirement of translation system, the non-parametric metric learning module without any learning

parameters is used to calculate similarity value. Here, cosine similarity calculation strategy is adopted to calculate the similarity value between the predicted translation result and the real label, and the model is optimized end to end according to the similarity value. Furthermore, to reduce the difference between model-generated translation and real translation results, parametric metric learning is adopted [21]. The model is optimized in an end-to-end approach using the translation generated of the model with the real translation.

In the metric learning module, the LSTM network is used to embed the real translated sentences into the sentence-vector space. Then, the cosine similarity value between the corresponding generated translation and the real translation embedded vector is calculated; finally, the similarity between them is judged according to the similarity value. The similarity calculation diagram is shown in Figure 7. When the angle is greater than 90, that is, the two vectors are orthogonal to each other, indicating that the sentences are not related to each other. The cosine similarity between two sentences is calculated as shown in formula.

$$L_{\text{sim}} = \frac{S_1 \cdot S_2}{\|S_1\|\|S_2\|} = \frac{\sum_{i=1}^{n} S_{1_i}S_{2_i}}{\sqrt{\sum_{i=1}^{n}\left(S_{1_i}\right)^2}\sqrt{\sum_{i=1}^{n}\left(S_{2_i}\right)^2}}, \tag{12}$$

where S_1 and S_2 represent the dot products of the two sentences.

3. Experimental Results

3.1. *Construction of Big Data Corpus.* The mainstream Corpus of Contemporary American English (COCA),

---

Input: Initialize weights W_{xi} and W_{hi}, encoder features
(1) Using formula (7) to compute the forgetting gate feature vector ft;
(2) Using formula (6) to compute the input gate eigenvector i_t;
(3) Using formula (9) and formula (10) to calculate the input modulation gate characteristic vector g_t, and update c_{t-1} to c_t;
(4) Using formula (8) to calculate the output gate characteristic vector o_t;
(5) Repeat steps (2) and (4) until all feature decoding is completed, and the input of the last decoding unit is the output result of the decoder.
Output Decoder Result

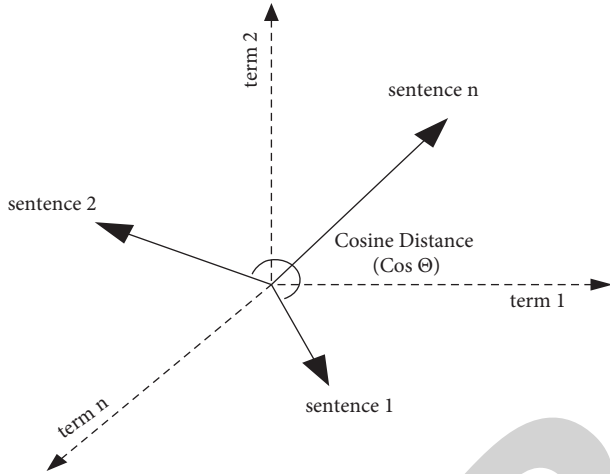ALGORITHM 1: Translation Generation Module.

FIGURE 7: Cosine similarity between sentences.

Michigan Corpus of Academic Spoken English (MICASE), and British National Corpus (BNC) are used to construct the big data corpus in this paper, and duplicate words are removed, which consist of 3.8 million words. It contains business lecture, seminars, consultation meeting, business negotiations, etc. And a translation result of the Youdao is selected to construct a word label. In addition, we use the results of Youdao translation as additional labels. In the nonparametric metric learning phase, the model is optimized in an end-to-end approach with additional labels. According to the ratio of 7 : 3, all data are divided into the training set and testing set.

3.2. Experimental Environment. We implement the proposed approach using PyTorch, and extensive experiments are implemented on 256 GB CPU, NVIDIA A100 GPU with 40 GB memory. CUDA environment is adopted NVIDIA CUDA 11.3 and cuDNN V8.2.1 deep learning acceleration library, and *Python* version is 3.5.2. We set the weight decay of 0.0005, initial learning rate of 0.0001, and reduced by a factor 0.1 after every 0.0004.

Figure 8 shows the accuracy and loss curves for the training and testing phases of the proposed model. From the analysis of the accuracy and loss curves in the training and testing phases, it can be seen that when the model epoch is 35, the accuracy and loss curves in both the training and testing sets tend to stabilize, and then the

curves change slightly, indicating that the model has converged. Therefore, the iteration epoch of all experiments in this paper is set as 35.

3.3. Evaluation Indicators. Accuracy, false-negative rate (FNR), false-positive rate (FPR), and F1-score are adopted as evaluation metrics in this paper, and all of them are shown in formula (13). Table 1 gives the confusion matrix of the classification. Generally, the higher the accuracy and F1-score, the lower the FPR and FNR, indicating the better the model performance. In addition, in order to avoid the unstable result of a single experiment, the average value of five experiments is used as the final evaluation result.

$$
\begin{aligned}
\text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN}, \\
\text{FPR} &= \frac{FP}{TN + FP}, \\
\text{FNR} &= \frac{FN}{TN + FN}, \\
F1 - \text{score} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.
\end{aligned}
\tag{13}
$$

4. Results and Analysis

The comparison experiments are conducted on the same data set so as to validate the performance of our model. Here, the model provided in literature [22–25] is used as comparative approaches. And the above methods are represented as A, B, C, and D, respectively. The results are presented in Table 2.

It can be seen that the overall advantage of this model is more obvious compared with the current mainstream translation query model. Particularly, for the accuracy, compared with the two best performing models A and B in all comparison models, this model is improved by 0.81% (94.08%⟶94.85%) and 1.33% (93.60%⟶ 94.85%), respectively. On the FNR side, compared with the two best performing models A and B in all comparison models, in this paper, the model decreased by 6.53% (5.05%⟶4.72%) and 10.78% (5.29%⟶4.72%), respectively. On the FPR, compared with the two best performing models A and C in all comparison models, the proposed model decreases by 10.29% (4.86%→4.36%) and 13.32% (5.03%→4.36%)
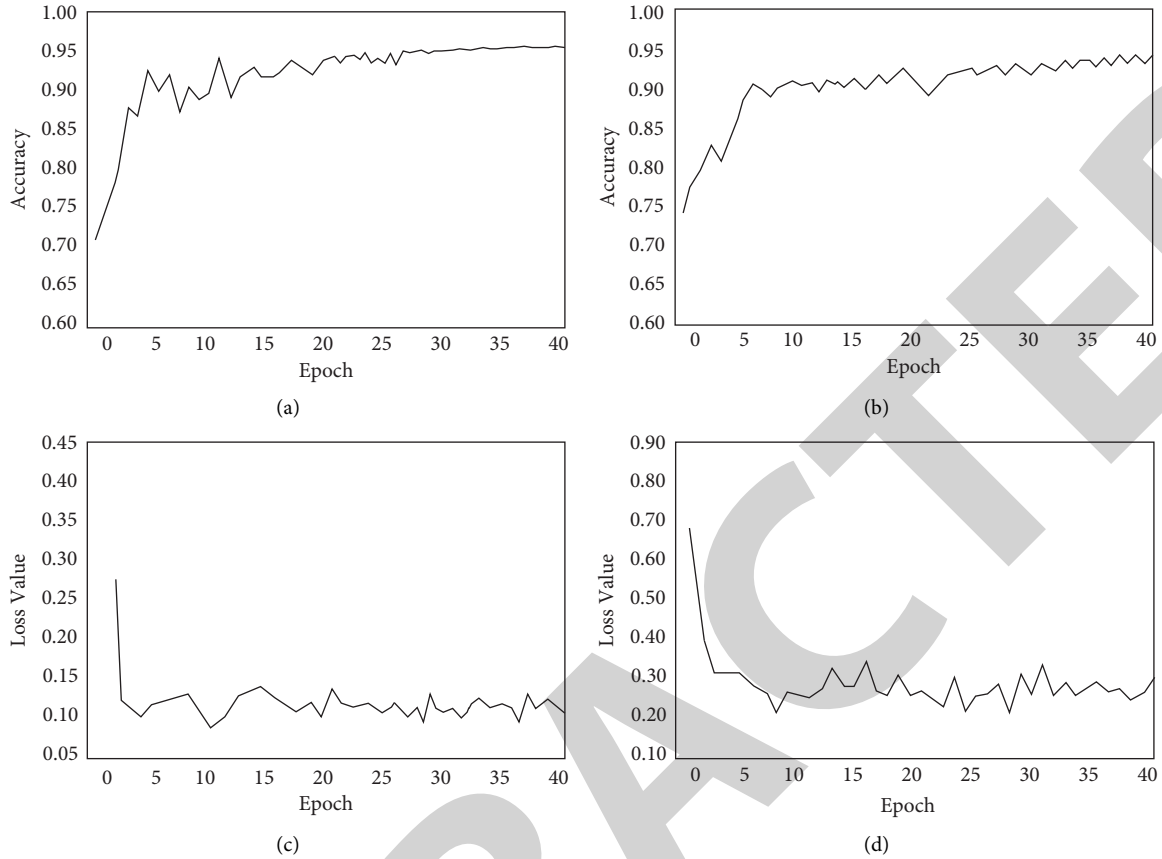
FIGURE 8: Accuracy and loss curves on training and validation sets. (a)Accuracy curves of the training set. (b) Verification of the accuracy curve of the set. (c) Training set loss curve. (d) Verify the set loss curve.

TABLE 1: Confusion matrix.

The actual class	Predict class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

TABLE 2: Experimental results of different models (%).

Methods	Accuracy	FNR	FPR	F1
A	94.08	5.05	4.86	94.38
B	93.60	5.29	5.18	93.82
C	92.91	6.01	5.03	93.22
D	91.79	6.52	7.14	92.05
Ours	94.85	4.72	4.36	95.01

FIGURE 9: Real-time comparison results of different models.

respectively. On the F1 side, compared with the two best performing models A and B in all comparison models, this model is improved by 0.66% (94.38%⟶95.01%) and 12.68% (93.82%⟶95.01%), respectively. The experimental results above show that the overall advantage of the model is obvious under all evaluation metrics.

Besides, Figure 9 shows the results of the proposed model compared with the mainstream A, B, C, and D models mentioned above in terms of time overhead (TO). It is clear that the proposed model has better real-time
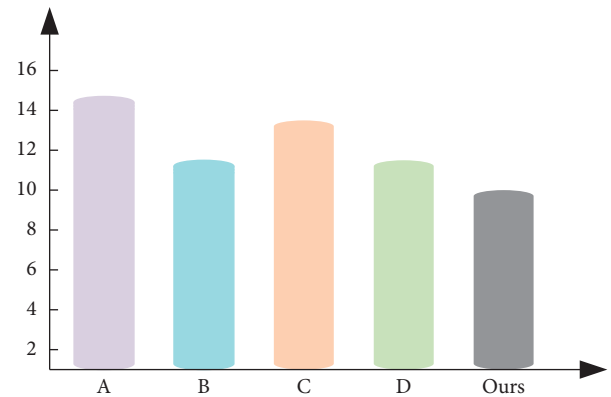
performance while maintaining translation and query accuracy.

4.1. Ablation Studies. In order to analyze the effects of deeply separable convolution, attention mechanism, and each module on the overall model translation query performance, an ablation experiment was conducted. Figure 9 shows the influence of each module on the overall model performance. The ablation experiment is conducted to analyze the influence of the deep-separable convolution, attention

Table 3: Ablation results.

No.	CNN	DSC	AM	NPM	Accuracy (%)	FNR (%)	FPR (%)	F1 (%)	TO (s)
1	√		√	√	94.91	4.69	4.32	95.03	18.60
2		√		√	91.20	7.95	8.04	91.80	9.85
3		√	√		94.37	4.80	4.53	94.32	11.61
4		√	√	√	94.85	4.72	4.36	95.01	10.39

mechanism, and metrics-free learning module on the overall model translation query performance. Detailed results are presented in Table 3, and CNN represents that only CNN is used as the backbone network to map the input statements to the deep feature space. DSC represents depth-separable convolution; AM stands for attention mechanism; NPM stands for nonparametric metric learning module.

According to the experiments in No. 1 and No.4 in Table 3, compared with the traditional CNN as a feature extractor, the depth-separable convolution is used to map the input statements into the depth feature space. Although there is a decrease in accuracy, F1, FNR, FPR, and other evaluation indicators, the decrease is not significant. However, the model translation query time decreases by 44.13% (18.6s⟶10.39s). It shows that the depth-separable convolution introduced in this paper has a positive effect on the reduction of model inference time. According to the experiments of No. 2 and No. 4, although the inference time of AM model introduced with the attention mechanism decreased, accuracy decreased by 3.85% (94.85%⟶91.20%). Therefore, the attention mechanism has a great impact on the overall performance of the model. According to the experiments of No. 3 and No. 4, nonparametric metric learning module has better real-time performance than traditional training mode because there are no learnable parameters and can also improve the overall performance to a certain extent.

5. Conclusion

This paper proposes a query and support decision model for English translation with the big data corpus and applies it to business English translation. Using deep-separable convolution as feature extractor instead of traditional convolutional network can effectively reduce the time overhead. Secondly, the attentional mechanism is introduced into the proposed model to further optimize the performance with strengthening the expression ability of the depth features. Finally, a nonparametric metric learning module is used to improve the model in an end-to-end approach. Compared with all comparison models, the proposed model performs well in translation and query performance and has better real-time performance on the built big data corpus.

In the future work, the proposed model can be deployed to PC and mobile clients to provide timely and convenient translation and query services.

Data Availability

The experimental data used to support the results of this study can be obtained from the author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest in this work.

References

[1] D. Diefenbach, V. Lopez, K. Singh, and P. Maret, "Core techniques of question answering systems over knowledge bases: a survey," *Knowledge and Information Systems*, vol. 55, no. 3, pp. 529–569, 2018.

[2] J. H. Chen, M. C. Su, V. T. Azzizi, T. K. Wang, and W. J. Lin, "Smart project management: interactive platform using natural language processing technology," *Applied Sciences*, vol. 11, no. 4, p. 1597, 2021.

[3] A. Casey, E. Davidson, M. Poon et al., "A systematic review of natural language processing applied to radiology reports," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 1–18, 2021.

[4] M. Ahirrao, Y. Joshi, and A. Gandhe, "Phrase Composing Tool Using Natural Language processing," in *Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1–4, Hubli India, June 2021.

[5] J. Paladines and J. Ramirez, "A systematic literature review of intelligent tutoring systems with dialogue in natural language," *IEEE Access*, vol. 8, pp. 164246–164267, 2020.

[6] R. Triguero, "Machine translation systems and quality assessment: a systematic review," *Language Resources and Evaluation*, vol. 56, no. 2, pp. 1–27, 2021.

[7] A. Barbosa, M. Ferreira, and M. Ferreira, "The Impact of Automatic Text Translation on Classification of Online Discussions for Social and Cognitive presences," in *Proceedings of the The 11th International Learning Analytics and Knowledge Conference*, pp. 77–87, Irvine CA USA, April 2021.

[8] B. Savoldi, M. Gaido, L. Bentivogli, M. Negri, and M. Turchi, "Gender bias in machine translation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 845–874, 2021.

[9] J. Cheng, J. Liu, and X. Xu, "A review of Chinese named entity recognition," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 6, pp. 2012–2030, 2021.

[10] M. Moradshahi, V. Tsai, and G. Campagna, "Contextual semantic parsing for multilingual task-oriented dialogues," 2021, https://arxiv.org/abs/2111.02574.

[11] J. Kumar, S. Henge, and P. Dubey, "An intensive literature review on machine translation using neural approach," *Recent Innovations in Computing*, pp. 45–59, Springer, Berlin, Germany, 2022.

[12] G. Zhang, Y. Dong, and H. Chen, "Double-channel multilayer information fusion for text matching," in *Proceedings of the 2022 International Conference on Intelligent Information Processing*, pp. 114–123, Springer, Foshan China, February 2022.

[13] X. Li, H. Zang, X. Yu et al., "On improving knowledge graph facilitated simple question answering system," *Neural Computing & Applications*, vol. 33, no. 16, pp. 10587–10596, 2021.

[14] P. Li and K. Mao, "Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts," *Expert Systems with Applications*, vol. 115, pp. 512–523, 2019.

[15] P. Liu, W. Yuan, and J. Fu, *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language processing*, 2021,https://arxiv.org/abs/2107.13586.

[16] Z. Y. Khan and Z. Niu, "CNN with depthwise separable convolutions and combined kernels for rating prediction," *Expert Systems with Applications*, vol. 170, Article ID 114528, 2021.

[17] B. Hua, M. Tran, and S. Yeung, "Pointwise convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 984–993, San Juan, PR, USA, June 2018.

[18] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[19] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[20] X. Bai, "Text Classification Based on LSTM and attention," in *Proceedings of the 2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pp. 29–32, IEEE, Berlin, Germany, September 2018.

[21] R. C. Belwal, S. Rai, and A. Gupta, "Text summarization using topic-based vector space model and semantic measure," *Information Processing & Management*, vol. 58, no. 3, Article ID 102536, 2021.

[22] Q. Xie, X. Zhou, J. Wang, X. Gao, X. Chen, and L. Chun, "Matching real-world facilities to building information modeling data using natural language processing," *IEEE Access*, vol. 7, pp. 119465–119475, 2019.

[23] P. Sonawane, K. Shah, and P. Patel, "Speech to indian sign language (ISL) translation system," in *Proceedings of the 2021 international conference on computing, communication, and intelligent systems (ICCCIS)*, pp. 92–96, IEEE, Greater Noida, India, February 2021.

[24] B. Pan and Q. Qin, "Construction of parallel corpus for English translation teaching based on computer aided translation software," *Computer-Aided Design and Applications*, vol. 19, no. S1, pp. 70–80, 2021.

[25] D. K. Jain, R. Jain, Y. Upadhyay, A. Kathuria, and X. Lan, "Deep Refinement: capsule network with attention mechanism-based system for text classification," *Neural Computing & Applications*, vol. 32, no. 7, pp. 1839–1856, 2020.