IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 December 2023, accepted 11 January 2024, date of publication 15 January 2024, date of current version 24 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3354723

RESEARCH ARTICLE

Hierarchical and Multiple-Perspective Interaction Network for Long Text Matching

ZHUOZHANG ZOU[1,2,3,4], ZHIDAN HUANG[5], WEI YANG[1,2,3,4], LONGLONG PANG[1,2,3,4], CHEN LIANG[1,2,3,4], AND QUANGAO LIU[1,2,3,4]

[1]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
[2]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[3]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[4]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[5]School of Mechanical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

Corresponding author: Wei Yang (epicard@163.com)

This work was supported by the Natural Science Foundation of Gansu Province under Grant 21JR11RA067.

ABSTRACT Long text matching is widely used in various sub-tasks of natural language processing. However, conducting research in this field can be challenging due to excessive redundant and distracting information, the complex semantic context, and the limited availability of high quality public datasets. Existing long text matching methods generally do not fully use the rich local features embedded in text information, and focus more on encoding long text as fixed length vectors to calculate the semantic distance, disregarding the importance of feature interaction in the text matching process. Therefore, the performance of the relevant models needs to be improved. To address these problems, a hierarchical and multiple-perspective interaction network (HMIN) is proposed in this paper. First, the long text is encoded at the word and sentence levels to extract global features, while one-dimensional convolutional neural networks and attention mechanisms are used to focus on important local features in long texts. Second, the different types of features are compared separately using the comparison function, and then, the comparison results are aggregated. Finally, whether long texts are matched is determined in the prediction layer. We have conducted comparative experiments on two datasets, the results show that HMIN has an improvement in accuracy and F1 values compared with the same type of existing algorithms, and the related experimental analysis demonstrates the effectiveness of the proposed method.

INDEX TERMS Long text matching, hierarchical attention, local features, interaction network.

I. INTRODUCTION

Text matching is a classical and important problem in the field of natural language processing, which is widely used in question and answer systems, information retrieval, recommendation systems [1], [2], and machine reading comprehension [3], [4]. The text matching process determines the matching relationship between the two texts by analyzing the semantics of source and target texts. The format of the source and target texts used for matching varies greatly from job to job, and it can be classified as short-to-short text matching, short-to-long text matching, long-to-short text matching, and long-to-long text matching based on the length of the text. For example, in the question answering system based on frequently-asked questions, the source and target texts are two problems of relatively short length, and short-to-short text matching is used to find multiple target problems with the same semantics as the source problem [5]. In the information retrieval task, the source text is a relatively short length query condition, the target text is a relatively long length document content, and short-to-long text matching is used to determine whether the document content matches the query condition [6]. In sentiment classification work, the source text is a relatively long length paragraph document and the target text is a relatively short length sentiment category, but this is a special form of matching, which is more often implemented as text classification [7], [8], [9], [10], [11]. In news content recognition work, the source and target texts

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Huang.

are both news documents of relatively long length, and the semantic content is analyzed through long text matching to determine whether the two news articles describe the same event or story [12]. Usually, different methods are required for matching texts of different lengths. Although many excellent results have been achieved in the research of matching related to short texts, the lack of publicly available high-quality datasets and the requirement of high resource allocation for training have led to insufficient research and relatively few excellent algorithmic models for long text matching. With the explosive growth of paragraph- and document-level text data and the increasing demand for long text matching tasks in related work, long text matching has gradually become an active research topic in the field of natural language processing [13].

There are several major challenges with long text matching. First, in long texts words or phrases have more ambiguous and diverse semantics, and the same word or phrase in the context may have multiple meanings. Therefore, how to effectively model the semantics of long texts is one of the challenges. Second, long texts have richer information content and complex text structure, the number of words or phrases in long texts range from hundreds to thousands, and they may contain multiple sentences, paragraphs, and even various levels of headings. Therefore, how to use rich information and complex structure to mine the true semantics of words or phrases and the key semantics is one of the challenges. Finally, among the neural network-based text matching methods, the encode-based method that encodes text as a fixed length vector is simple, convenient, and general. However, interactive text matching methods that consider word alignment or comparison between text pairs usually exhibit better performance [14]. Therefore, choosing an appropriate method to effectively interact between features in the process of long text matching is one of the challenges.

For the challenge that the same word or phrase may have multiple meanings, pre-trained language models represented by BERT can already solve this problem well by generating dynamic representation vectors, and can show better performance compared to non-pre-trained models [15]. However, at the same time, pre-trained language models have the limitation of input length, which also generates a large computational overhead in the training process, especially when the text length is long [16]. In this study, we focus on the second and third challenges in the long text matching process, and therefore do not discuss much about the pre-trained-based model, or compare the proposed model with it.

Many studies have been conducted to extract more important semantic information from texts using attention mechanism methods [17], [18], and these methods can achieve good results when the sequence length is relatively short. However, as the length of the text sequence increases, the semantic information and global true semantics with important values will gradually be diluted or even hidden in the long text sequence. Therefore, it is crucial to effectively use the self-contained hierarchical structure and rich information content in long texts to understand the true semantics and focus on important semantic information at relatively short sequences of different levels. Some representative approaches are to carry out the operations of attention mechanisms on long texts in word and sentence sequences separately, focusing on more important semantic contents in different levels while disassembling the text structure, and finally aggregate the results to obtain the global true semantics of long texts [19], [20]. Jiang et al. [21] further developed this by considering the paragraph level of long texts, linking different levels of representation to capture the features of different hierarchical structures. However, although some existing studies have made progress in addressing this challenge, they still do not consider the rich information content of long texts. Failing to consider the richer information content that contains more local features does not affect the short text related matching process because the number of local features contained in short texts is limited. However, ignoring the richer information content that contains more local features greatly affects the long text matching process, where a large number of local features can be of great help in understanding the important semantics and true semantics of the full text as well.

Additionally, the interaction of text pairs is an important task in text matching because interactive text matching models usually perform better on relevant datasets. In particular, various variants of algorithms based on the attention mechanism have been widely applied to the interaction operations between text pairs, and impressive milestones have been achieved by computing the weight matrix by semantically aligning words or phrases in the current text with all words or phrases in another text [22], [23]. Furthermore, some studies have combined the attention mechanism of soft alignment or the attention mechanism of hard alignment with various computational methods such as dot product, cosine similarity, and Euclidean distance to perform a comparison between text pairs after semantic alignment of words or phrases to capture difference features and similarity features [24], [25], [26]. However, these interaction methods use attention mechanisms for semantic alignment and are not universal in the process of long text matching, because the computation process of obtaining the attention weight matrices of two texts generates a huge computational overhead, thus limiting the application scenarios of the algorithms. Therefore, it is sometimes difficult to apply the attention mechanism directly for the semantic alignment of words or phrases in long texts. Some studies do not use the attention mechanism in the interaction process of text matching, but use methods such as coupled-LSTMs or multi-perspective matching [27], [28], [29]. Although there is no semantic alignment of words or phrases, the interaction of comparing text pairs achieves the same good results.

In this paper, we propose a hierarchical and multiple-perspective interaction network (HMIN) to address the

second and third challenges of the matching process for long texts. First, HMIN uses hierarchical attention to process long text sequences with complex text structures, and uses one-dimensional (1D) convolutional neural networks to focus on local features in long texts, which effectively exploits the rich information content in long texts and extracts important semantics and true semantics in long texts by combining them with global features obtained through a multi-level attention mechanism. Second, HMIN is an interactive long text matching model. Although no attention mechanism is used for semantic alignment between words and phrases, HMIN uses a combination of correspondence position difference, correspondence position product, and neural network to perform coarse-grained interaction between text pairs to achieve attention to similarity features and difference features between features of the same type in long texts. Finally, HMIN combines the different comparison results to predict the matching results for long texts. The main contributions of this study can be summarized as follows:

1) We propose an HMIN for long text matching, which constructs an interactive model from hierarchical and multiple-perspective perspectives to achieve the prediction of matching results between long texts.

2) We make full use of the complex text structure and rich information content in long text and use hierarchical attention to capture the global features of long texts and 1D convolutional neural networks to capture the local features of long texts. Furthermore, we combine the two to focus on the important semantics and true semantics of long text, and demonstrate the importance of the large number of local features contained in long text for long text matching.

3) Unlike most previous long text matching models, the proposed text matching model interactively compares global and local features between text pairs, verifying that even coarse-grained comparison methods without semantic alignment of words or phrases are helpful for model performance improvement in the process of text matching.

The rest of this paper is organized as follows. We introduce text matching and related works on long text matching in Section II. The problem statement and the explanation of the model structure are given in Section III. In Section IV, we present and analyze the experimental results of the model on relevant datasets. In Section V, we present the conclusions of this study. In Section VI, we present an outlook on future work.

II. RELATED WORKS

A. TEXT MATCHING

Determining the matching relationship between different text semantics become an active research topic in the field of natural language processing, and therefore, many models have been proposed to determine the matching relationship. These models can be broadly classified into three categories: shallow feature-based approaches, deep learning-based approaches and pre-training-based models. The shallow feature-based approach uses the terminology of the text, the syntactic structure, and the WordNet dictionary to determine the degree of match between texts. Examples include TF-IDF, which uses the word frequency of terms and the inverse document frequency to calculate the degree of text matching; BM25 [30], which further considers information such as text length and average text length; and LDA [31] and LSA [32], which analyze the potential topics of the text. Li and Li [33] used the syntactic structure in the text to calculate the similarity, thereby overcoming the shortcomings of traditional methods, which could not handle the complex grammar in the text. Tsatsaronis et al. [34] used the constructed word corpus to calculate the degree of matching between texts. Although these matching methods seem to be more intuitive and easier to understand, they always lack effective analysis of the semantics embedded in the text, and the development of deep learning provides an opportunity to solve this problem. Deep learning-based methods can learn hidden semantic features on large scale public datasets, and they can be classified into encode-based methods and interactive methods according to their means of implementation. The encode-based approach is to encode the text into fixed length feature vectors and, subsequently, calculate the semantic distance between the feature vectors. Neculoiu et al. [35] used Siamese-structured BiLSTM to extract global features of the text, and model the semantics of the text, and subsequently used similarity scores to measure the degree of text matching with good results. Yin et al. [22] used CNN to extract local features of the text and obtained feature vectors for prediction results using different pooling and splicing strategies. Wang et al. [36] instead combined RNN and CNN to encode the text, focusing on both global and local features of the text, and the combined model outperformed both the traditional RNN and CNN models on relevant datasets. The interactive approach is to predict the result of text matching after deep interaction of features from different texts. Wang et al. [37] proposed a compare-aggregate method in which after one-way semantic alignment, the text is compared using multiple comparison functions, followed by feature aggregation using 1D convolutional neural networks, and finally, the results are predicted. Chen et al. [38] instead performed a bidirectional semantic alignment and then compared the semantic features of different texts using a difference and product calculation to achieve interaction between texts. Hu et al. [24] proposed a context-aware cross-attention mechanism that focuses on contextual information during interaction to achieve better semantic alignment. In contrast, Wang et al. [28] did not perform semantic alignment of words or phrases but used full-matching, maxpooling-matching, attentive-matching, and max-attentive-matching to achieve interaction between texts, and several studies have demonstrated the exceptional performance of this multi- perspective matching mechanism [39]. Pre-training based approaches using transformer structure for supervised or unsupervised training on

large scale data, represented by BERT and RoBERTa, have achieved very excellent results in the field of text matching. BERT separates different sentences using [SEP] token and inputs them into a regression function to make judgments, and achieves advanced performance. RoBERTa achieves better performance compared to BERT by modifying the pre-training process [40]. Liu et al. proposed Sentence-BERT based on BERT, which reduces the computational overhead and inference time of the model and makes the pre-trained language model more suitable for text matching tasks [41].

B. LONG TEXT MATCHING

Most of the existing text matching methods are not designed for long texts, so there are inevitably some problems in applying these methods directly to long text matching tasks. For example, the RNN model has difficulty capturing long term dependencies in long texts, and important information is lost in the process of passing information in long sequences. Although the LSTM and gating unit can effectively solve this problem, there is still the problem of gradient disappearance or gradient explosion. In addition, the long text has a huge computational overhead for the semantic alignment of words or phrases using the attention mechanism due to its long sequence.

Chen et al. [40] improved the DSSM algorithm based on BiGRU and DAttention to capture the semantic information in long texts and reduce the interference of noise and redundant information. This approach inherits essentially the idea of short text matching and does not effectively use the unique and complex text structure features in long texts. Some studies, however, have encoded long texts from the perspective of text structure using hierarchical properties in the text. For example, Yang et al. [19] used hierarchical attention to model the representation features at the word and sentence levels of long texts, and they aggregated the representation features at the sentence level to obtain the representation features of the whole long text. Chen et al. [20] similarly obtained sentence level representation features based on word level representation features and then obtained long text representation features based on sentence level representation features to consider user and product information through hierarchical attention. Jiang et al. [21] focused on the paragraph level in long texts in addition to the word and sentence levels and aggregated the representation features at different levels to learn the representation features of long texts. Due to the effective use of multiple levels of text structure, the model can learn semantic features of different depths, thus allowing better modeling of long texts, and therefore such methods are widely used in tasks such as long text matching and long text classification. This approach of hierarchical attention reflects the understanding process of humans, i.e., people tend to understand the semantics of words or phrases when reading long texts, further understand the meaning of entire sentences, then understand the overall semantics of the content under the heading of a paragraph or a level, and finally gain an understanding of the semantics of long texts by summarizing it at the full-text level. However, almost all of these studies construct a Siamese-structured neural network and use similarity measures to measure the degree of text matching after modeling, always lacking the operation of interacting features of different long texts, and there is still room for further exploration. Some studies modeled long texts from the perspective of graph neural networks. For example, Liu et al. [12] used concept interaction graphs to represent long texts as concept graphs and later evaluated whether the long texts matched with each other after comparison and aggregation operations with very good results. There are also some studies that use BERT for long text matching from a pre-training point of view. However, the input length of BERT is limited to 512, and the text exceeding the length will be truncated. Sun et al. [43] proposed different truncation methods to enable BERT to process long texts, but this inevitably results in semantic loss. While Yang et al. [44] proposed siamese multi-depth transformer-based hierarchical to extend the input length to 2048. In addition, Gan et al. [45] proposed TBNF based on transformer to improve the performance by reducing the noisy data in long text. Pang et al. [46] proposed Match-Ignition based on transformer, which also facilitates the matching process for long texts by reducing noisy data. All these pre-training based models have achieved more leading rankings on relevant datasets, but they all require huge computational overheads, especially in the context of long text matching aggravating the burden of model training.

III. MATH

A. PROBLEM STATEMENT

The long text matching task is similar to the common short text matching task in that both tasks compute whether the semantics expressed by the two are the same given a text pair $T = \{T_a, T_b\}$. Unlike short text matching, which gives two relatively short sentences, long text matching usually gives two longer documents that contain more words, sentences, paragraphs, and headings. So the task can be described as giving a source document T_a and a set of candidate documents $T = \{T_1, T_2, T_3, \ldots\ldots, T_n\}$, the text matching model needs to evaluate the semantic similarity $y = sim\{T_a, T_i\}, 1 \leq i \leq n$, so that the source and target documents can obtain higher similarity scores [21]. Depending on how the similarity results are computed, the process of long text matching can be considered a classification task or a regression task, and in this study, we consider it a binary classification task that predicts semantic matches or mismatches.

B. FRAMEWORK OVERVIEW

Figure 1 illustrates the framework of our proposed HMIN. First, the text sequences of long texts are converted into

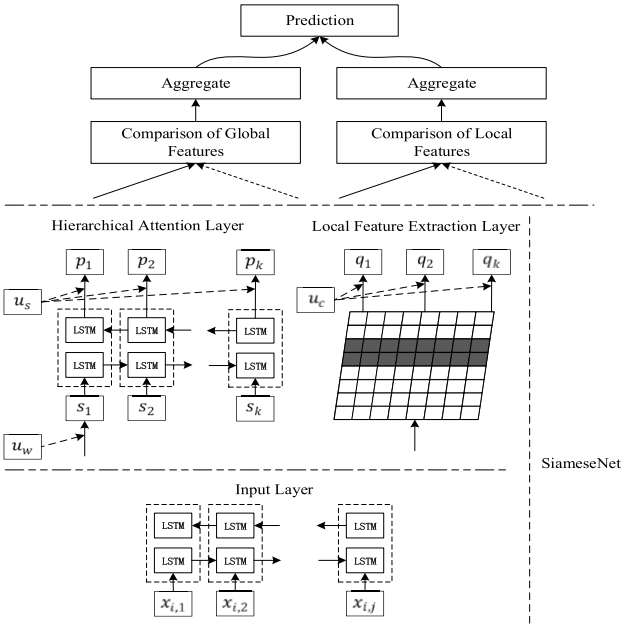


FIGURE 1. Overview of HMIN.

representation vectors with certain associations by Word2Vec in the input layer, and the feature vectors are obtained by encoding the word level representation vectors using BiLSTM. Next, the word level feature vectors are fed into the Siamese structure of the hierarchical attention layer as well as the local feature extraction layer. In the hierarchical attention layer, in this study, we use the attention mechanism and BiLSTM to focus on the more important contents of feature vectors at different levels and encode the sentence level representation vectors as feature vectors. In the local feature extraction layer, we use 1D convolutional neural networks to extract local features between words of a specific length and also use an attention mechanism to obtain locally focused features. Then, the two different types of features interact in depth using the comparison function separately, and the features after the interaction operation are aggregated using 1D convolutional neural networks. Finally, the outputs of the comparison aggregation layer are combined in the prediction layer, and the matching result of long texts is predicted by the fully connected layer.

In this study, ''hierarchical,'' ''multiple-perspective,'' and ''interaction network'' are three important concepts.

1) The ''hierarchical'' refers to the fact that the proposed model use both word and sentence level information in a long text in the hierarchical attention layer, and uses different levels of attention mechanisms for different levels of information to focus on the more important content. Furthermore, ''hierarchical'' can include not only the word and sentence levels but also paragraph, heading, or other levels with clearly delineated boundaries, which is a optional configuration. Because many long Chinese texts have a complex, redundant, and specialized semantic environment, and the text quality is difficult to guarantee, the use of word separation tools for Chinese long texts may bring uncertain results, which in turn affects the accuracy of long text matching models. Therefore, the word level in HMIN can be replaced with the character level in the context of long Chinese text matching.

2) The ''multiple-perspective'' refers to the fact that the proposed model models long text from a global perspective in the hierarchical attention layer and focuses on the features between characters of a specific length in each sentence of long texts from a local perspective in the local feature extraction layer. Furthermore, it uses an attention mechanism to focus on the more important parts of the extracted local features. Because long texts have richer information contents than short texts, and some finer-grained features are inevitably overlooked in the global modeling of long texts, it is necessary to extract features from multiple perspectives for local information of long texts.

3) The ''interaction network'' refers to the fact that instead of using a similarity function such as a cosine function to predict the result based on two fixed length vectors obtained from the Siamese structure, HMIN uses a combination of corresponding position difference, corresponding position product, and nonlinear neural network functions to interact the two vectors with coarse-grained features. Furthermore, HMIN aggregates the interacted features to predict the result of long text matching.

C. HIERARCHICAL AND MULTIPLE-PERSPECTIVE INTERACTION NETWORK

1) SEQUENCE ENCODER

LSTM conveys sequence information through the gating mechanism and cell state, which can solve the problems such as information forgetting that exist in RNN when long sequence input is used. Figure 2 shows its cell structure.

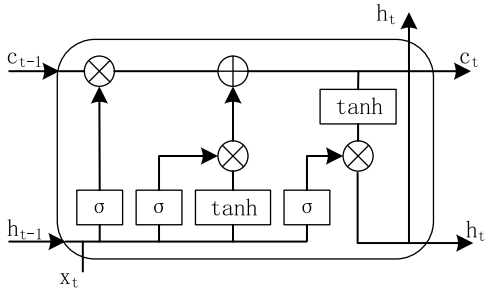


FIGURE 2. LSTM unit structure diagram.

Each LSTM cell contains input gates, forgetting gates, output gates, and cell states, which together determine the memory or forgetting of the information, control the updating of cell states and pass them to the next moment in the cell. Given the input x_t at the current moment t, the hidden state h_t of the LSTM can be calculated using the following equations:

$$f_t = \sigma\left(W_f\left[x_t, h_{t-1}\right] + b_f\right) \tag{1}$$

$$O_t = \sigma\left(W_O\left[x_t, h_{t-1}\right] + b_O\right) \tag{2}$$

$$i_t = \sigma\left(W_i\left[x_t, h_{t-1}\right] + b_i\right) \tag{3}$$
$$\tilde{C}_t = tan\,h\left(W_C\left[x_t, h_{t-1}\right] + b_C\right) \tag{4}$$
$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1} \tag{5}$$
$$h_t = O_t * tanhC_t \tag{6}$$

where f_t, O_t and i_t represent the states of the forgetting output, and input gates at moment t, respectively; W_f, W_O, and W_i represent the weight matrices of the respective corresponding gate structures; b_f, b_O, and b_i represent the bias values of the respective corresponding gate structures; σ is the sigmoid function used in the gate structure calculation process; and $\tilde{C}_t$ and C_t are the candidate and cell states at moment t, respectively. The state of the LSTM hidden layer at the current moment is obtained by passing the cell state to the *tanh* function and multiplying it with the output gate state.

To obtain the information features of long text in both directions (forward and backward) at the same time, BiLSTM is used as the encoder of the sequence in this study. BiLSTM can obtain the output result $\vec{h}_t$ of the LSTM unit hiding layer in the forward direction and the output result $\overleftarrow{h}_t$ of the LSTM unit hiding in the reverse direction at moment t, and $h_t = \left[\vec{h}_t, \overleftarrow{h}_t\right]$ obtained by splicing the two is the output result of the BiLSTM hiding layer at moment t.

2) INPUT LAYER

Given the long texts a and b to be matched in the input layer, in this study, we use $T_{i,j,l}$ to denote the jth character in the ith sentence of a long text, and the length of the sentence is l; that is, the ith sentence contains a total of l Chinese characters. However, in the set of candidate documents mentioned in the task description, different long texts have different numbers of sentences, and different sentences in the same long text have different numbers of characters, i.e., long texts have different sequence lengths in different levels, and there may be relatively large differences in lengths between sequences in the same level. Therefore, to ensure memory-friendly and facilitate the training of the model, in this study, we specify the length of a document as the number of sentences in the document and the length of a sentence as the number of characters in the sentence. Furthermore, we denote the sequence length of a set of candidate documents as k and the sequence length of sentences as l. If the length of a sentence in a long text is less than l, we ensure that the length of the sentence is l by merging multiple sentences or using special token padding. Furthermore, if the length of a sentence is greater than l, the sentence is truncated at length l. Similarly, if the length of a document in a set of candidate documents is less than k, a fixed length sentence filled with special token is used to fill the current document, and if the length of a document is greater than k, the current document is truncated at length k. After such data processing, we can obtain a set of long text sequences with the same document length and sentence length in the document. Based on this, we start to encode and extract features from long texts.

In this study, we use Word2Vec as the embedding model for character vectors and convert the text sequence of long texts into a vector sequence that can be computed and processed by finding the corresponding character representation vectors. The character level representation vector obtained by embedding the jth character $T_{i,j,l}$ in the ith sentence of the current long text with Word2Vec is denoted by $x_{i,j}$. Its formula is shown as follows:

$$x_{i,j} = Word2Vec\left(T_{i,j,l}\right) \tag{7}$$

To obtain the character level feature vectors, we use BiLSTM as a character level encoder to obtain the forward and backward features of characters at each position. BiLSTM outputs the forward and backward hidden layer results $\vec{h}_{ij}$ and $\overleftarrow{h}_{ij}$, respectively, as shown in Equations (8) and Equation (9), respectively. By combining these equations, we obtain $h_{i,j}$, which is the character level feature vector that summarizes both the information forward and backward of the sentence at position $x_{i,j}$. Finally, the output of the character level encoder $\left\{h_{i,1}, h_{i,2} \ldots\ldots h_{i,l}\right\}$ is obtained by combining the feature vectors of all characters in a sentence.

$$\overrightarrow{h_{ij}} = \overrightarrow{BiLSTM}\left(x_{i,j}\right) \tag{8}$$
$$\overleftarrow{h_{ij}} = \overleftarrow{BiLSTM}\left(x_{i,j}\right) \tag{9}$$
$$h_{i,j} = \left[\overrightarrow{h_{ij}}, \overleftarrow{h_{ij}}\right] \tag{10}$$

3) HIERARCHICAL ATTENTION LAYER

In this layer, we are concerned with encoding long text from a hierarchical perspective and extracting its global features. Therefore, we divide a long text into character and sentence levels, obtain the sentence level representation vector from the character level feature vector from the bottom to the top, and then extend it to the extraction of the sentence level features. In addition, different attention mechanisms are used at different levels to focus on more important information features at different levels. Figure 3 shows the hierarchical attention layer.

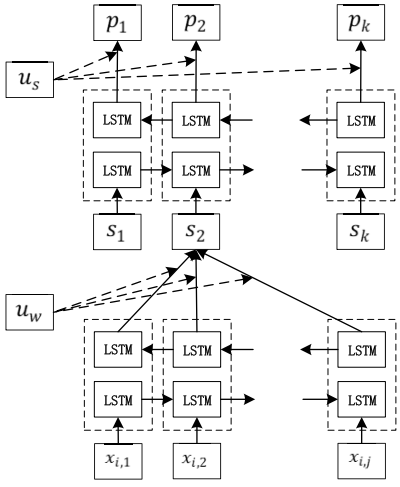


FIGURE 3. Hierarchical attention operation.

Usually, not every character in a sentence has the same importance weight for the semantic expression of the current sentence, or some characters do not even contribute to the semantic expression of the current sentence. Therefore, we need to use a character level attention mechanism to find which character's feature vector has a greater importance weight for constructing the semantics of the current sentence. In addition, after distinguishing the importance of different characters to the semantic expression, the feature vectors of each character in a sentence are aggregated separately to form the representation vector of the current sentence. The formulas of the character level attention mechanism are as follows:

$$u_{i,j} = tanh(W_w h_{i,j} + b_w) \tag{11}$$

$$a_{i,j} = \frac{exp(u_{i,j}^T u_w)}{\sum_j exp(u_{i,j}^T u_w)} \tag{12}$$

$$s_i = \sum_j a_{i,j} h_{i,j} \tag{13}$$

where W_w, b_w, and u_w are the weights to be learned by the character level attention mechanism. We first use a multilayer perceptron to obtain the hidden representation $u_{i,j}$ of $h_{i,j}$, then use a randomly initialized vector matrix u_w as the character level context vector of the current sentence to calculate the importance of the current character, and use the obtained importance weights to obtain the attention weight matrix $a_{i,j}$ using the softmax function to perform a normalized calculation. Finally, the character level feature vectors are recalculated using Equation (13) based on the obtained character level importance weights, and the new character level feature vectors are further aggregated to obtain the representation vector s_i of the current sentence semantics.

Given the representation vector s_i of a sentence, we also use the sequence encoder at the sentence level to obtain the sentence level feature vector with the following equations:

$$\vec{h}_i = \overrightarrow{BiLSTM}(s_i) \tag{14}$$

$$\overleftarrow{h}_i = \overleftarrow{BiLSTM}(s_i) \tag{15}$$

$$h_i = \left[\vec{h}_i, \overleftarrow{h}_i\right] \tag{16}$$

where h_i is the feature vector of sentences encoded at position i that summarizes the information forward and backward the long text. The output information of the sentence level encoder $\{h_1, h_2 \ldots\ldots h_k\}$ can be obtained by combining the feature vectors of all sentences in the long text.

Similarly, not all sentences in a long text can play an equally important role in the text matching process, so we continue to introduce sentence level attention mechanism operations to measure the importance weights of different sentences:

$$u_i = tanh(W_s h_i + b_s) \tag{17}$$

$$a_i = \frac{exp(u_i^T u_s)}{\sum_i exp(u_i^T u_s)} \tag{18}$$

$$p_i = a_i h_i \tag{19}$$

We obtain the hidden representation u_i of h_i by the multilayer perceptron, where W_s, b_s are the weight matrix to be learned in multilayer perception. A randomly initialized vector matrix u_s is used as the current long text sentence level context vector, and the attention weight matrix a_i is obtained after normalized calculation using the softmax function. However, here we do not aggregate all the sentence feature vectors in a long text to obtain the document representation vector as done by Yang et al. [19] because they focused more on using different hierarchical structures to construct the document representation vector and using predefined similarity functions or a fully concatenated layer to predict the semantic match. However, many experiments have demonstrated that interactive text matching models can achieve better accuracy and F1 values than encode-based text matching models. Therefore, we use the compare and aggregate layers to further compare the vector representation of different sentences in the subsequent operation of the comparison function to determine the relationship between two long texts through richer similarity features or different features. Furthermore, the sentence level feature vector of long texts obtained after the hierarchical attention layer can be expressed as follows:

$$p = \{p_1, p_2, \ldots\ldots, p_k\} \tag{20}$$

4) LOCAL FEATURE EXTRACTION LAYER

A long text contains global features that can semantically represent and generalize its features and local features that focus on specific step length characters or words, which also play an indispensable role in the process of long text matching. In this paper, we use 1D convolutional neural networks to extract these local features between characters, and Figure 4 shows their structure.

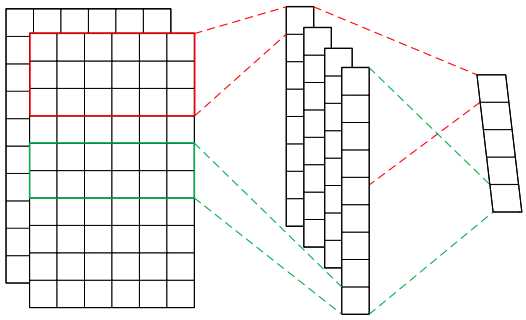


FIGURE 4. Hierarchical attention operation.

In the input layer, we obtain the character level feature vector $h_{i,j}$ and the ith sentence of length l in the long text $\{h_{i,1}, h_{i,2} \ldots\ldots h_{i,l}\}$. One-dimensional convolutional neural networks use a convolutional kernel W_c, which is applied to a character window of a specific length to obtain the local features of this sentence. Unlike the two-dimensional convolutional neural network applied to the vision domain, the convolutional kernel of 1D convolutional neural networks has a dimension that is fixed, i.e., its length is the same as the

dimension of the feature vector. In this paper, the convolution kernel $W_c \in D \times L$, where L is the length of the convolution kernel, which is equal to the dimension of the output of the encoder BiLSTM unit, and D is the height of the convolution kernel. The formula for the 1D convolutional neural networks is as follows:

$$g_j = f\left(W_c h_{i,j:j+D-1} + b_c\right) \tag{21}$$

where b_c is the bias term of the 1D convolutional neural networks, f is a nonlinear function, and $h_{i,j:j+D}$ represents the feature vector of characters in the window at each operation. As the convolutional kernel keeps sliding according to a specific step, the corresponding feature map is obtained by applying Equation (21) to the different character intervals $\{h_{i,1:D}, h_{i,2:D+1} \ldots\ldots h_{i,l-D+1:l}\}$ of the current sentence as follows:

$$g = \{g_1, g_2, \ldots\ldots, g_{l-D+1}\} \tag{22}$$

As with the hierarchical attention layer, different local features in a sentence have different importance weights, so we also use the attention mechanism for the local features obtained from the 1D convolutional neural networks to focus on the more important parts of them, as shown in the following equations:

$$u_{cnn,i} = tanh(W_c g_i + b_c) \tag{23}$$

$$\beta_i = \frac{exp(u_{cnn,i}^T u_c)}{\sum_i exp(u_{cnn,i}^T u_c)} \tag{24}$$

$$q_i = \beta_i g_i \tag{25}$$

where W_c, b_c, and u_c are the weights to be learned by the local feature attention mechanism. After the local feature vector obtained from the 1D convolutional neural network is input to the multilayer perceptron, the hidden representation $u_{cnn,i}$ can be obtained, which is normalized using the softmax function together with the randomly initialized context vector u_c to calculate the weight matrix β_i of the attention mechanism. Furthermore, $u_{cnn,i}$ is multiplied with the local feature vector to obtain the new local feature vector, which is a better representation of the more important local features in the current sentence compared with the output of the 1D convolutional neural network.

5) COMPARE AND AGGREGATE LAYERS

Through hierarchical and multiple-perspective calculations, we can obtain the global feature outputs p_a and p_b of the source and target documents, and the local feature outputs q_a and q_b of the source and target documents. In order to achieve interaction among different features, we use a comparison function that combines multiple methods. The comparison function includes the corresponding position product, corresponding position difference, and neural network operation. Additionally, this comparison function focuses on the similarity between different features while also focusing on the difference between different features, so it has an excellent comparison effect. Equations (26) and (27) show the comparison functions of global and local features, respectively:

$$f_h = Relu(W_h \begin{bmatrix} (p_{ai} - p_{bi}) * (p_{ai} - p_{bi}) \\ p_{ai} * p_{bi} \end{bmatrix} + b_h) \tag{26}$$

$$f_q = Relu(W_q \begin{bmatrix} (q_{ai} - q_{bi}) * (q_{ai} - q_{bi}) \\ q_{ai} * q_{bi} \end{bmatrix} + b_q) \tag{27}$$

where f_h is the comparison function used for global features; W_h and b_h are the weight matrix and bias vector of the global feature comparison function, respectively. Furthermore, f_q is the comparison function used for local features, and W_q and b_q are the weight matrix and bias vector of the local feature comparison function, respectively. We use multiplication operations in the process of corresponding position difference operations to avoid causing negative numbers.

We use 1D convolutional neural networks as an aggregation method to extract the features after the comparison function operation. The results of the global feature comparison function operation and the results of the local feature comparison function operation are aggregated separately, and the formulas are as follows:

$$v_p = f\left(W_p f_{h,i:i+D-1} + b_p\right) \tag{28}$$

$$v_q = f\left(W_q f_{q,j:j+D-1} + b_q\right) \tag{29}$$

where W_p and W_q are two filters, b_p and b_q are bias terms of 1D convolutional neural networks, and f is a nonlinear function.

6) PREDICTION LAYER

After obtaining the outputs of the hierarchical global and step-specific local features in the compare and aggregate layers, we splice the two together v to obtain a feature vector v for predicting the matching results of the long-form text. After inputting this feature vector into the fully connected layer, the softmax function is used to determine whether the two long-form texts match each other. The equation of the prediction layer is as follows:

$$v = \{v_p, v_q\} \tag{30}$$

$$P = softmax\left(W_o v + b_o\right) \tag{31}$$

where W_o and b_o are the weight and bias vectors to be learned in the fully connected layer, respectively.

We use the cross-entropy function as the loss function for model training, which is given in the following equation:

$$loss = -\left(y log\hat{y} + (1-y)\, log\left(1-\hat{y}\right)\right) \tag{32}$$

where y is the true label value; the values of the positive and negative categories are 1 and 0, which represent the semantic match and semantic mismatch between source and target documents, respectively. Additionally, $\hat{y}$ is the probability value obtained from the model prediction, and $\hat{y}$ is greater than or equal to 0 and less than or equal to 1. The cross entropy

function reflects the difference between the model prediction and the real sample results.

IV. RESULT

A. DATASETS

We conducted experiments on two publicly available datasets for the long document matching task: the Chinese News With Events (CNSE) dataset and the Chinese News With Stories (CNSS) dataset. Both datasets are taken from major Chinese Internet news providers, and their contents cover a variety of topics in the open domain, and are constructed by professional editors after tagging high-quality long-form Chinese news articles. The average number of characters in these long-form Chinese news articles is 734, and the maximum number of characters is 21,791. A total of 29,063 sample pairs are included in the CNSE dataset, with positive and negative samples of 12,865 and 16,198, respectively, while 33,503 sample pairs are included in the CNSS dataset, with positive and negative samples of 16,887 and 16,616, respectively.

In both datasets, the two long Chinese texts in the positive sample describe the same breaking news, and the two texts in the negative sample are not simply randomly combined; they are generated after considering similar keywords and TF-IDF similarity. We divide the two datasets as done by Liu et al. [12]: 60% of the total data are used as the training set, 20% as the validation set, and the remaining 20% as the test set while ensuring no data overlap or omission. Furthermore, the accuracy rate and F1 value of binary classification are selected as evaluation metrics, and the evaluation metric formulas are shown as follows

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{33}$$

$$pre = \frac{TP}{TP + FP} \tag{34}$$

$$recall = \frac{TP}{TP + FN} \tag{35}$$

$$F1 = \frac{2 * pre * recall}{pre + recall} \tag{36}$$

where TP denotes the number of samples correctly predicted as text matching, TN denotes the number of samples correctly predicted as text mismatching, FP denotes the number of samples incorrectly predicted as text matching, and FN denotes the number of samples incorrectly predicted as text mismatching. We can calculate the specific values of acc and F1 through the performance results of the model in the test set.

It is worth noting that the authors of the original datasets provided not only high-quality long-form articles but also corresponding headlines and keywords within the articles and performed accurate word separation operations on the long-form Chinese news articles. However, such refined headlines, keywords, and accurate word separation results are not often available in real world applications or other large-scale public datasets, and most of the text matching datasets only provide a pair of texts and a label. To ensure the ease and applicability of our proposed model to other scenarios or datasets, we do not use the titles, keywords, and word separation results from the CNSE and CNSS datasets, but we only extract the long texts and divide them by characters.

B. TRAINING DETAILS

In the input layer of the model, we use Word2Vec character embedding vectors of dimension 300, and the characters outside the word list are set to the same random vector. In the input and hierarchical attention layers, we use a one-channel convolutional neural network with a kernel height of 3, a width of 128, and a step size of 1. In the comparison aggregation layer, we use a three-channel CNN with a kernel height of 3, a width of 256, and a step size of 1. The model is trained using the Adam optimizer to update the training parameters and set the training batch size to 100. The model is trained using batch normalization to prevent gradient explosion or gradient disappearance and using dropout to prevent overfitting. The model is trained on a 16 core Intel Xeon Platinum 8369B CPU, with 120 G of memory and an A100 GPU.

C. EXPERIMENTAL RESULTS

To verify the effectiveness of our proposed model, we compared it with existing popular long text matching models, including term-based similarity measures, encoded neural network models, and interactive neural network models.

1) Term-based traditional methods (BM25 and LDA): The experimental results of these term-based unsupervised learning methods depend on the accurate word separation of the CNSE and CNSS datasets.

2) Encoding-based neural network model (ARC-I, DSSM, C-DSSM, MASH-RNN, and HAN_Attention): These methods extract text features and encode them into fixed-length vectors to calculate the semantic distance between them.

3) Interaction-based neural network model (ARC-II, MatchPyramid, DUET, and HAN_Interaction): These methods consider feature interactions between different texts during text matching. HAN_Interaction is a model for coarse-grained interaction based on HAN-Attention using the same comparison function as in this paper.

Tables 1 and 2 show the experimental and comparison results of the above baseline and HMIN models on the CNSE and CNSS datasets, respectively.

As can be seen in Tables 1 and 2, the HMIN model achieved 72.90% accuracy and 70.28% F1 value in the CNSE dataset, and 71.27% accuracy and 72.01% F1 value in the CNSS dataset. Therefore, the HMIN model performed better on both datasets compared with the baseline models. This can be due to the following two reasons. First, the HMIN model considers the rich local features embedded in long

TABLE 1. Experimental and comparison results of the baseline and HMIN models on the CNSE dataset.

Model	acc (%)	F1(%)
ARC-I	53.84	48.68
ARC-II	54.37	36.77
DUET	55.63	51.94
DSSM	58.08	64.68
C-DSSM	60.17	48.57
MatchPyramid	66.36	54.01
MASH-RNN	62.13	56.92
LDA	63.81	62.44
HAN_Attention	64.40	60.69
BM25	69.63	66.60
HAN_Interaction	71.15	67.83
HMIN	**72.90**	**70.28**

TABLE 2. Experimental and comparison results of the baseline and HMIN models on the CNSS dataset.

Model	acc (%)	F1(%)
ARC-I	50.10	66.58
ARC-II	52.00	52.83
DUET	52.33	60.67
DSSM	61.09	70.58
C-DSSM	52.96	56.75
MatchPyramid	62.52	64.56
LDA	62.98	69.11
MASH-RNN	64.31	64.48
BM25	67.77	70.04
HAN_Attention	68.22	68.53
HAN_Interaction	70.16	71.16
HMIN	**71.27**	**72.01**

texts, which helps understand the important semantics and true semantics in long texts. Therefore, the HMIN model improves its accuracy and F1 value on the CNSE dataset by 1.75% and 2.45%, respectively, and also improves its accuracy and F1 value on the CNSS dataset by 1.11% and 0.85%, respectively, compared with the HAN_Interaction model that does not consider the rich local features. Second, the HMIN model performed interaction operations on the extracted features, respectively, and although these are coarse-grained interaction operations, the accuracy and F1 values of HMIN are better than those of HAN_Attention of the encoding method, improving the accuracy and F1 values by 8.5% and 9.59% on the CNSE dataset, respectively, and also improves its accuracy and F1 value on the CNSS dataset by 3.05% and 3.48%, respectively. In addition, the experimental results show that the methods that consider hierarchical attributes can usually achieve better accuracy and F1 values compared with those that do not consider hierarchical attributes, which laterally verifies that the effective use of multi-level features in long texts is important to help extract semantic features in long texts.

V. CONCLUSION

In this study, we have proposed an HMIN, which extracts global features of long texts by encoding the text at different levels and using an attention mechanism to focus on important information. At the same time, a 1D convolutional neural network is used to focus on the rich local features in the long text, and the attention mechanism is used to focus on the important information. Finally, the different types of features are compared separately and aggregated to predict the results of long text matching. Experiments on a large scale public dataset demonstrate that the proposed model outperforms existing popular long text matching methods. The experimental results and correlation analysis show that the rich local features are of great help in understanding the true and important semantics of long texts and that the coarse-grained interaction method using only comparison functions is indispensable for long text matching, which can bring some improvement to the model performance.

VI. FUTURE WORK

The proposed HMIN model still has much room for improvement in terms of accuracy and F1 values, and there are two main factors. First, in this paper, we did not use pre-trained language models, for example, BERT. Such language models are pre-trained with a combination of supervised and unsupervised data on a large scale; therefore, they can provide better text representation and can solve the problem of ambiguity in semantics in text very well. Second, we did not use attention mechanisms for semantic alignment to achieve finer-grained interactions in the process of text matching. By calculating the importance of words in one long text based on all position words in another long text one by one, we can identify semantic links between texts and better model the relationship between text pairs. However, both approaches incur huge computational overhead during training and, therefore, need to be analyzed in the process of practical application to determine whether they should be used according to the specific situation.

REFERENCES

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, Apr. 2017, pp. 173–182.
[2] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surveys*, vol. 52, no. 1, pp. 1–38, Jan. 2020, doi: 10.1145/3285029.
[3] S. Wang and J. Jiang, "Machine comprehension using match-LSTM and answer pointer," 2016, *arXiv:1608.07905*.
[4] A. Wei Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "QANet: Combining local convolution with global self-attention for reading comprehension," 2018, *arXiv:1804.09541*.
[5] Z. Wu, J. Liang, Z. Zhang, and J. Lei, "Exploration of text matching methods in Chinese disease Q&A systems: A method using ensemble based on BERT and boosted tree models," *J. Biomed. Informat.*, vol. 115, Mar. 2021, Art. no. 103683.

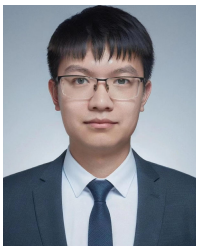
[6] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for Web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.-(CIKM)*, San Francisco, CA, USA, 2013, pp. 2333–2338.
[7] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, 2015, pp. 1422–1432.
[8] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscipl. Rev. Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1253, Jul. 2018.
[9] A. Onan, "Hierarchical graph-based text classification framework with contextual node embedding and BERT-based dynamic fusion," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 7, Jul. 2023, Art. no. 101610.
[10] A. Onan, "Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2098–2117, May 2022.
[11] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Syst. Appl.*, vol. 57, pp. 232–247, Sep. 2016.
[12] B. Liu, D. Niu, H. Wei, J. Lin, Y. He, K. Lai, and Y. Xu, "Matching article pairs with graphical decomposition and convolutions," 2018, *arXiv:1802.07459*.
[13] J. Chen and S. Lv, "Long text truncation algorithm based on label embedding in text classification," *Appl. Sci.*, vol. 12, no. 19, p. 9874, Sep. 2022.
[14] W. Lan and W. Xu, "Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering," in *Proc. 27th Int. Conf. Comput. Linguistics*, Santa Fe, NM, USA, Aug. 2018, pp. 3645–3656.
[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[16] Y. Cheng, R. Chen, X. Yuan, Y. Yang, S. Jiang, and B. Yang, "Overview of long-form document matching: Survey of existing models and their challenges," *J. Phys. Conf. Ser.*, vol. 2171, no. 1, Jan. 2022, Art. no. 012059.
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 1–55.
[18] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, 2016, pp. 606–615.
[19] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, San Diego, CA, USA, 2016, pp. 1480–1489.
[20] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, 2016, pp. 1650–1659.
[21] J.-Y. Jiang, M. Zhang, C. Li, M. Bendersky, N. Golbandi, and M. Najork, "Semantic text matching for long-form documents," in *Proc. World Wide Web Conf.*, San Francisco, CA, USA, May 2019, pp. 795–806.
[22] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 259–272, Dec. 2016.
[23] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom, "Reasoning about entailment with neural attention," 2015, *arXiv:1509.06664*.
[24] Z. Hu, Z. Fu, Y. Yin, and G. de Melo, "Context-aware interaction network for question matching," 2021, *arXiv:2104.08451*.
[25] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," 2016, *arXiv:1606.01933*.
[26] H. He and J. Lin, "Pairwise word interaction modeling with deep neural networks for semantic similarity measurement," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, San Diego, CA, USA, 2016, pp. 937–948.
[27] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, 2015, pp. 1576–1586.
[28] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," 2017, *arXiv:1702.03814*.
[29] P. Liu, X. Qiu, and X. Huang, "Modelling interaction of sentence pair with coupled-LSTMs," 2016, *arXiv:1605.05573*.
[30] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends®Inf. Retr.*, vol. 3, no. 4, pp. 333–389 2009.
[31] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[32] T. G. Kolda and D. P. O'Leary, "A semidiscrete matrix decomposition for latent semantic indexing information retrieval," *ACM Trans. Inf. Syst.*, vol. 16, no. 4, pp. 322–346, Oct. 1998.
[33] X. Li and Q. Li, "Calculation of sentence semantic similarity based on syntactic structure," *Math. Problems Eng.*, vol. 2015, pp. 1–8, Jan. 2015.
[34] G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis, "Text relatedness based on a word thesaurus," *J. Artif. Intell. Res.*, vol. 37, pp. 1–39, Jan. 2010.
[35] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with Siamese recurrent networks," in *Proc. 1st Workshop Represent. Learn. NLP*, Berlin, Germany, 2016, pp. 148–157.
[36] C. Wang, F. Jiang, and H. Yang, "A hybrid framework for text modeling with convolutional RNN," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, Canada, Aug. 2017, pp. 2061–2069.
[37] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," 2016, *arXiv:1611.01747*.
[38] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," 2016, *arXiv:1609.06038*.
[39] D. Peng, S. Wu, and C. Liu, "MPSC: A multiple-perspective semantics-crossover model for matching sentences," *IEEE Access*, vol. 7, pp. 61320–61330, 2019.
[40] S. Chen and T. Xu, "Long text QA matching model based on BiGRU–DAttention–DSSM," *Mathematics*, vol. 9, no. 10, p. 1129, 2021.
[41] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[42] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.
[43] Z. Xinxi, "Single task fine-tune BERT for text classification," in *Proc. 2nd Int. Conf. Comput. Vis., Image, Deep Learn.*, Kunming, China, Oct. 2021, pp. 194–206.
[44] L. Yang, M. Zhang, C. Li, M. Bendersky, and M. Najork, "Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1725–1734.
[45] L. Gan, L. Hu, X. Tan, and X. Du, "TBNF: A transformer-based noise filtering method for Chinese long-form text matching," *Appl. Intell.*, pp. 1–15, Jun. 2023.
[46] L. Pang, Y. Lan, and X. Cheng, "Match-ignition: Plugging PageRank into transformer for long-form text matching," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 1396–1405.

ZHUOZHANG ZOU was born in Changchun, Jilin, China, in 1998. He received the master's degree from the Shenyang Automation Research Institute, in 2023. His main research interest includes natural language processing.

ZHIDAN HUANG was born in Jingyuan, Gansu, China. She received the degree from Lanzhou Jiaotong University. Her main research interests include machine learning and system dynamics analysis.

WEI YANG received the degree from the Department of Electronic Engineering, Fudan University, in 1991, and the master's degree in pattern recognition and intelligent systems from the Shenyang Institute of Automation, Chinese Academy of Sciences, in 1997. He is currently a Master's Tutor and a Researcher with the University of Chinese Academy of Sciences. His main research interest includes public health big data analysis.

LONGLONG PANG was born in Anhui, China, in 1993. He received the degree from Lanzhou Jiaotong University. He is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences. His main research interests include public health big data analysis and machine learning.



CHEN LIANG was born in Guangxi, China, in 1998. He received the degree from North China Electric Power University. He is currently pursuing the master's degree with the Shenyang Institute of Automation, Chinese Academy of Sciences. His main research interest includes computer vision.

QUANGAO LIU was born in Yibin, Sichuan, in 2000. He received the degree from Southwest Jiaotong University. He is currently pursuing the master's degree with the Shenyang Institute of Automation, Chinese Academy of Sciences. His main research interest includes public health big data analysis.