Received April 2, 2020, accepted April 16, 2020, date of publication April 20, 2020, date of current version May 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988918

Combining Local and Global Features Into a Siamese Network for Sentence Similarity

YULONG LI, DONG ZHOU, AND WENYU ZHAO
School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

Corresponding author: Dong Zhou (dongzhou1979@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876062, in part by the Hunan Provincial Innovation Foundation for Postgraduate under Grant CX2018B671, and in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 18B199.

ABSTRACT Sentence similarity is widely used in various natural language tasks such as natural language inference, paraphrase identification, and question answering. However, a variety of linguistic expressions and ambiguities of words in sentences make it difficult to measure sentence similarity. Many studies show that using local features or global features of a sentence will produce satisfactory sentence representations that can be utilized to measure sentence similarity. Local features reflect the relationships of adjacent words for each sentence and the sequence information of a sentence are usually expressed by global features. However, local features lack abilities to capture sequence information while a small amount of extracted global features is not enough to produce sentence representations with good qualities. In this paper, we propose A Hybrid Model combining Local and Global features into a Siamese Network (HM-LGSN) for sentence similarity calculation. We first propose a new convolution neural network architecture called group convolution neural network to extract the most representative local features (or word semantic features). Then we combine these new features with pre-trained embeddings of words as input to the Bidirectional Gated Recurrent Units to extract global features of sentences. Finally, we select the global features to form sentence representations and calculate sentence similarity through Manhattan distance. The experimental results on SICK, MSRVID, STS-B datasets show that the accuracy of our proposed model is significantly improved by combining local features and global features.

INDEX TERMS Sentence similarity, semantic representation, siamese network, local feature, global feature.

I. INTRODUCTION

Sentence similarity is a challenging research task with applications in many Natural Language Processing (NLP) tasks, such as question answering [1], document summarization [2], and sentence generation [3]. Due to the variety of linguistic expressions and ambiguities of words in sentences, it is difficult to measure the semantic similarity between them [4]. Traditional approaches measure sentence similarity by using features extracted manually, which is very time-consuming [5]–[7]. Moreover, the extracted features are usually sparse and insufficient. As an alternative, neural network techniques are widely utilized to extract a sufficient number of features automatically [8]–[11].

The associate editor coordinating the review of this manuscript and approving it for publication was Fan-Hsun Tseng.

Some studies calculate sentence similarity by considering relationships between matching units of a sentence pair [9], [12], [13]. However, these approaches use words or phrases as matching units between sentences and ignore the effect of the whole sentence on the similarity calculation. Calculating sentence similarity based on sentence representations is another choice for researchers. By generating so-called sentence vectors, similarity can be measured in different ways, such as cosine distance. He *et al.* [14] use a Convolutional Neural Network (CNN) to extract features of sentences in different granularity and propose a new feature filtering algorithm to form final sentence representations. Mueller and Thyagarajan [15] combine Long and Short Time Memory (LSTM) to encode sentences and learn sentence representations based on a Siamese network.

In general, CNN is capable of extracting local features and LSTM can extract global features (or context features).

The above approaches extract either local or global features alone as the key features for sentence representations. For example in [14], when using local features as the key features, global features are completely neglected. On the contrary, these methods using global features of sentences [10], [15] fail to consider local features of sentences, resulting in the problem of insufficient features. There are also some studies improving the quality of global features for the generation of sentence representations [16]–[18]. However, these approaches filter global features by using different types of pooling schema such as max-pooling, mean pooling, and so on, resulting in the loss of more features in a sentence. There are few studies (for example [19]) that try to combine local and global features together for calculating sentence similarity. However, they only consider very limited features and the results are somewhat unsatisfactory.

In this paper, we propose a Hybrid Model combining Local and Global features into a Siamese Network (HM-LGSN) for sentence similarity calculation. We firstly propose a novel convolution neural network architecture called group convolution neural network (G-CNN) to extract local features and get two types of feature maps derived from a large number of convolution filters. A sentence feature map can be produced by applying convolution filters to different windows of words in the sentence. A word feature map is produced by applying multi-scale convolution filters to the same window of words based on an assumption. There are some studies explaining that the local features of words can reflect the semantic information of words to some extend [20]–[22]. To get the most representative local features for words, we perform pooling operation on word feature maps which consist of different types of local features instead of performing pooling operation on sentence feature maps as usual [23]–[25]. Then we combine local features with pre-trained word embeddings to form new word semantic features. Finally, we use the Bidirectional Gated Recurrent Unit (Bi-GRU) to extract global features of a sentence with the input of new word semantic features. In this way, final sentence representations will contain both local and global information of sentences. After obtaining sentence representations, we use Manhattan distance of two sentence representations to measure sentence similarity. Besides, our HM-LGSN model is built based on the Siamese network [10], [26], [27] to model the semantic relationships of sentence pairs.

We conduct thorough evaluations on three test sets from two SemEval[1] STS competitions. The results show that our HM-LGSN model achieves better performance than other state-of-the-art sentence similarity measuring models. The contributions of this paper are summarized as follows:

- We present a hybrid model based on a Siamese network for sentence similarity calculation. This model integrates local features (through G-CNN) and global features (through Bi-GRU) to produce better sentence representations. Our model can solve the problem of insufficient feature extraction in current sentence similarity models.
- We propose a novel convolution neural network architecture. This architecture uses multi-scale convolution filters to extract local features as word semantic features. To get the most valuable features for words, we perform pooling operation on local features for a word under different convolution filters which are considered as a group convolution. In this way, each word can get effective local features in a sentence, which is useful for generating sentence representations.
- We conduct extensive experiments to verify our proposed model in this paper. Compared with six state-of-the-art baseline models, results show that our proposed model for sentence similarity calculation with local and global features performs significantly better.

The remainder of this paper is organized as follows: Section II presents an overview of related work; Section III introduces the proposed model in detail. Section IV reports the results. The paper is concluded in Section V.

II. RELATED WORK

Sentence similarity is a long-standing research area that attracts the attention of a considerable amount of researchers [25], [28], [29]. Traditional approaches use explicit text features to learn semantic relationships between sentences. Madnani *et al.* [30] use the statistical information of sentences in a corpus and the categories of words to judge the semantic relationships. Fernando and Stevenson [31] propose a semantic similarity method based on the co-occurrence relationships of words between sentences. Das and Smith [32] use an artificial knowledge network like WordNet[2] which contains the entity-relationship between words to describe the interaction information of sentences. However, the explicit features are always sparse and insufficient, especially at the sentence level. Recently, with the development of deep learning techniques, words can be well represented by distributed features. And words as the basic semantic components of sentences play a crucial role in the semantic representation generation process of sentences. Thus, a lot of researchers preferred to measure sentence similarity with the help of word representations in deep neural network [10], [9], [29].

Sentence similarity methods based on deep neural networks mainly fall into two categories: interaction methods [4], [13], [33] and sentence modeling methods [12], [15], [34]. The interaction methods firstly construct an interaction matrix that corresponds to the relationships of matching units between sentences. Then they use CNN or the Recursive Neural Network (RNN) to capture interaction information from the interaction matrix. Finally, these methods integrate the interaction information into the neural network to get the sentence similarity scores. Wan *et al.* [13] use the distributed representation of words to construct a word-level interaction matrix, then integrate local interactions with spatial RNN

[1] http://alt.qcri.org/semeval/

[2] https://wordnet.princeton.edu/

and calculate the sentence similarity based on the spatial RNN's outputs. They also propose a deep architecture for semantic matching of sentences [35]. This architecture aggregates the interactions of multi-positional sentence representations to capture the contextualized local information in the matching process while multi-positional sentence representations are extracted by using Bi-LSTM at different positions. Pang *et al.* [33] use an approach called Hierarchical Convolution to extract rich matching patterns at different levels and construct three types of matrices for collecting the interaction relationships of semantic matching units with an inspiration of CNN's success in image recognition. He *et al.* [4] use Bi-directional Long Short-Term Memory (Bi-LSTM) to encode sentences for constructing three types of interaction matrix, then integrate local interactions by combining with deep CNN, finally measure sentence similarity using the outputs of the deep CNN. Although these methods can make use of the interaction information between sentences, they always consider word-level or phrase-level interaction information and ignore the interaction information of the whole sentence.

For sentence modeling methods, they use the neural network to encode the sentences, then select representative features of the sentence as sentence representations, finally calculate the similarity of sentence pairs according to the sentence representations. He *et al.* [14] use CNN to extract local features of the sentence in different levels and propose a feature filter method to form more representative sentence representations. Yin *et al.* [9] use multi-layer convolution to extract local features as representations of sentences and propose three attention schemes that integrate mutual influence between sentences into CNNs. These approaches take local features as sentence representation and ignore the global features in the sentence. However, the global features can reflect the sequence relationships between words which are crucial for constructing the semantic meaning of a sentence. Therefore, researchers also take global features to produce sentence representations. Tai *et al.* [36] propose a modified LSTM architecture that can capture both the context information and the grammatical information of sentences. They use the architecture to encode sentences for generating sentence representations. Kiros *et al.* [37] propose an unsupervised method that combines Bi-directional Gated Recurrent Unit (Bi-GRU) with Encoder-Decoder architecture to extract global features to yield sentence representations. Mueller and Thyagarajan [15] combine LSTM with Siamese network architecture for generating sentence representations. They firstly use LSTM to encode sentences for extracting global features, then select the most representative features as sentence representations, finally measure sentence similarity by calculating Manhattan distance between sentence representations. Subramanian *et al.* [38] propose a training method with multi-task datasets. They use Bi-GRU to extract global features and learn general sentence representations by using a modified objective function. However, these sentence modeling methods only consider extracting global features of sentences and ignore the local features. Actually, local features of sentences reflect the relationships of adjacent words which are important components of sentence semantics.

There are also some studies improving the quality of global features when encoding sentences. Conneau [34] uses Bi-LSTM to extract sentence global features and filter these features by performing max-pooling operation. Sentence representations are derived from the filtered features. Nie and Bansal [17] use stacked bidirectional LSTM to extract global features and perform row max-pooling operation to form sentence representations. Chen *et al.* [18] explore generalized pooling methods to enhance sentence embeddings and propose a vector-based multi-head attention method that includes the widely used max pooling, mean pooling, and scalar self-attention as special cases. These approaches filter global features by performing various pooling operations so that the global features will be partially lost in this process and cannot reflect enough semantic information of sentences.

Apart from considering local features and global features separately, there are a few studies that try to combine local and global features together for calculating sentence similarity. For example, Pontes *et al.* [19] use CNN and LSTM in a Siamese network for this very purpose. However, they only consider very limited features and the results are somewhat unsatisfactory. In their work, only one type of convolution filter is used for extracting local features. In the process of capturing global features, they merely use a one-way LSTM that considers forward information. This rather simplified structure may miss some important information, both locally and globally. In their experiments on the SICK dataset (also used in our experiments, together with two more datasets), they also find that only modest results can be obtained by their model.

In this paper, we propose a hybrid model for measuring sentence similarity. The model uses G-CNN to extract sufficient local features and integrates these features into the semantic features of words in a sentence, then encodes sentence pair with Bi-GRU based on the Siamese network architecture to extract global features. Finally, we select global features to produce sentence representations and calculate sentence similarity according to the above representations. The experimental results show that the proposed model improves the accuracy of sentence similarity.

III. AN HM-LGSN MODEL FOR SENTENCE SIMILARITY CALCULATION

Our HM-LGSN model for calculating sentence similarity consists of three components: (1) Multi-scale feature extraction. In this component, we use G-CNN to extract multi-scale local features in sentences which demonstrate relationships between locally continuous words or phrases. (2) Sentence encoding. We use Bi-GRU to encode global features and produce sentence representations in this component. To combine local and global features effectively, we integrate local features of words into the distributed features of these words and input the new word feature representations to Bi-GRU for

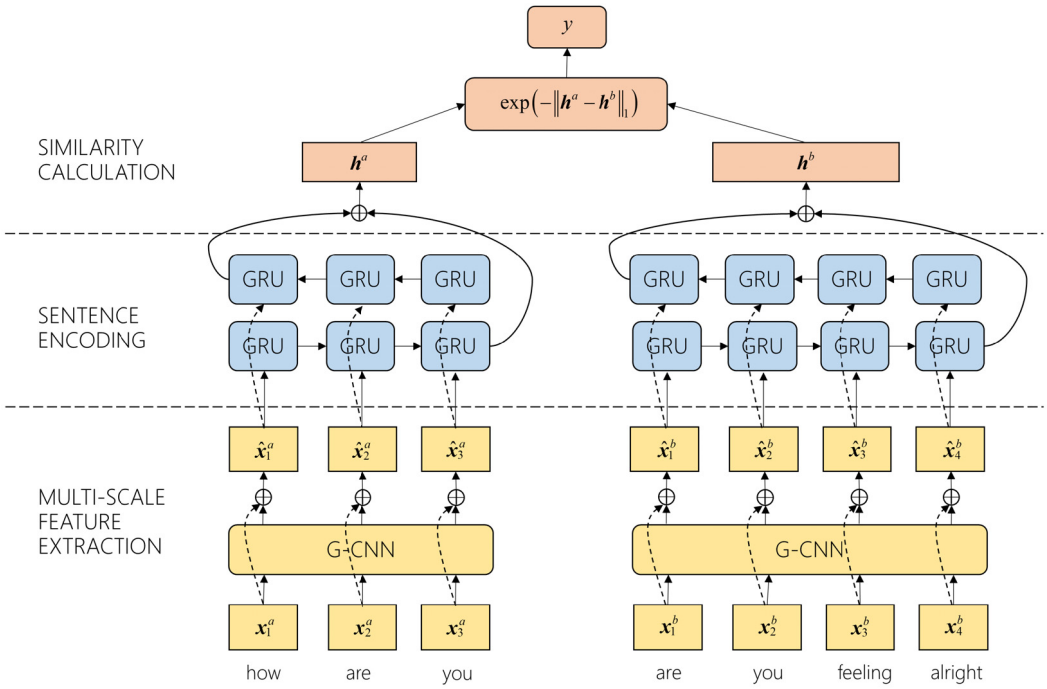

FIGURE 1. A HM-LGSN model for sentence similarity calculation.

learning global features and generating sentence representations. (3) Sentence calculation. After obtaining sentence representation, we use Manhattan distance to calculate sentence similarity in this component. Fig. 1 gives an illustration of our proposed HM-LGSN model.

A. MULTI-SCALE FEATURE EXTRACTION

Input to our HM-LGCN model is a pair of sentences $S^a = (x_1^a, x_2^a, \cdots, x_m^a)$ and $S^b = (x_1^b, x_2^b, \cdots, x_n^b)$, where x_i^a and x_i^b denotes d-dimensional vector of words in a sentence S^a and S^b respectively. Since a sentence is made up of several words and words cannot be directly operated by neural networks in their literal forms, we need to transfer them into embedding vectors before feature extraction.

CNN has been proved to be able to effectively extract local features in many NLP tasks [24], [39], [40]. When using CNN to encode sentences, local features of sentences are extracted by employing a convolution operation, then filtered through pooling operation. These features in multi-scale are extracted by applying convolution filters of different sizes. In this paper, we propose a new CNN architecture called group convolution neural network (G-CNN) to extract local features in sentences as word semantic features. Similar to traditional CNN, our G-CNN architecture includes a convolution layer and a pooling layer.

1) MULTI-SCALE CONVOLUTION

Multi-scale convolution is an approach to extract different types of local features in sentences. We first assume that local features extracted by the convolution filters can be used as the semantic feature of words, then the word that in the middle of the window under the convolution filter is chosen in our model. Based on this assumption, each filter is applied to every possible window of words in the sentence to get semantic features of words. By applying multi-scale convolution filters, different types of local features can be extracted.

As illustrated in Fig. 2, we define convolution filters of three different scales, respectively denoted as $k_1 \in \mathbb{R}^{3\times d}$, $k_2 \in \mathbb{R}^{5\times d}$ and $k_3 \in \mathbb{R}^{7\times d}$. d represents the dimension of pre-trained word embeddings. For example, for a sentence S^b, it is represented as follows:

$$\boldsymbol{x}_{1:n} = \boldsymbol{x}_1 \oplus \boldsymbol{x}_2 \oplus \cdots \oplus \boldsymbol{x}_n \tag{1}$$

where $\oplus$ is the concatenation operator, and $\boldsymbol{x}_{1:n}$ denotes a two-dimensional matrix consisting of a cascade of word vectors in the sentence. To ensure that each word can get the corresponding semantic features, we use the method of equal-width convolution [8]. Compared with other convolution methods [25], [41], the number of local features extracted by this method is consistent with the sentence length, which corresponds to the number of words in the sentence.

However, since we extract local features of the sentence rather than the image, the equal-width convolution is only performed in vertical (sequential) direction, not in horizontal direction. In this paper, we apply the equal-width convolution to each window of words in the sentence. This convolution method can capture n-gram features of sentences which play an important role in enriching sentence semantic. For instance, let a convolution filter denoted as k, and $k \in \mathbb{R}^{h\times d}$.

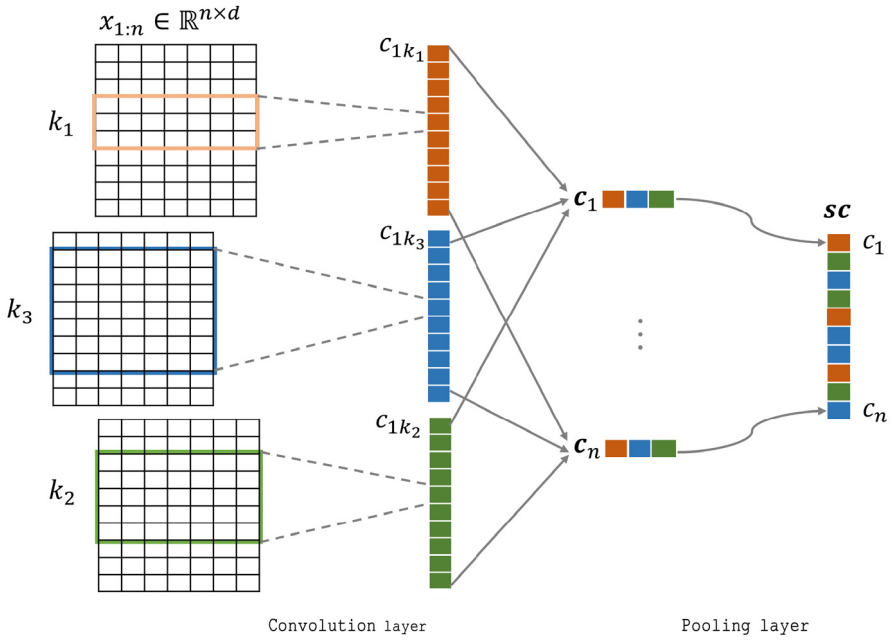

FIGURE 2. Group convolution neural network for an example sentence.

h represents the number of words under the convolution filter. Based on our assumption, we set h to odd numbers for extracting local features as semantic features of one word. Then the local feature value c_{ik} corresponding to the i-th word in a sentence is obtained through convolution operation as the following:

$$c_{ik} = f\left(w_k * x_{i-\frac{h-1}{2}:i+\frac{h-1}{2}} + b\right) \tag{2}$$

where w_k represents the weight matrices of the convolution filter k and b represents the bias term. $f(\cdot)$ represents a non-linear function, like hyperbolic tangent function.

2) POOLING STRATEGY

In the Pooling layer, compared with the previous pooling of sentence feature vectors [42]–[44], we propose a novel yet simple pooling strategy. There are two main differences in our strategy. One is the use of the group convolution filter. Since each word can get various semantic features corresponding to local features of sentences by applying different scales of convolution filters, we define these convolution filters as a group convolution filter to unify the local features for the word as a word feature map. This is a process of preparing for subsequent feature selection. Another is the horizontal pooling of local features for words. To select the most representative local features as word semantic features for each word, we perform pooling operations on the word feature map under a group convolution filter. The detailed steps are as follows:

Step 1. We combine the local features for each word under different scales of convolution filters to get a unified word feature map. The word feature map consists of some feature values extracted by performing convolution operation. For example, the word feature map $\boldsymbol{c}_i$ corresponds to the i-th word in a sentence is shown as following:

$$\boldsymbol{c}_i = \left[c_{ik_1}, c_{ik_2}, c_{ik_3}\right] \tag{3}$$

Step 2. We apply max pooling to the word feature map of each word, and thus we can select the most representative features for words under a group convolution filter. The max-pooling formula refers to (4),

$$c_i = max\left(\boldsymbol{c}_i\right) \tag{4}$$

where c_i denotes the most representative local feature value corresponding to the i-th word in a sentence.

Step 3. We apply one convolution filter to each window of words for generating a sentence feature map. The sentence feature map $\boldsymbol{sc}$ is shown as follows:

$$\boldsymbol{sc} = [c_1, c_2, \cdots, c_n] \tag{5}$$

Since one group convolution filter can only be applied to words in a sentence for generating a word feature map, we use multiple group convolution filters to extract more local features in a sentence. The last local feature vector of the i-th word which combines local feature value from all group convolution filters is shown as follows:

$$\boldsymbol{cw}_i = \left[c_i^1, c_i^2, \cdots, c_i^s\right] \tag{6}$$

where s denotes the number of group convolution filters and c_i^s represents the local feature value of the i-th word under the s-th group convolution filter.

B. SENTENCE ENCODING

Generally, sentence encoding is a process of transforming a sequence of word embeddings from a sentence into a fixed dimension vector. There are a lot of studies encoding sentences with CNN or the variants of RNN [45], [46]. CNN can extract local features effectively, while variants of RNN (eg. LSTM, GRU) are suitable for extracting global features. A sentence is composed of several words as an ordered sequence, thus global features are more effective to use for generating sentence representations. However, global features extracted from a sentence are insufficient. Therefore, we integrate local features into pre-trained word embeddings and extract global features by encoding a sentence which consists of new word semantic representations.

The main component of sentence encoding used in this paper is Bi-GRU. As a bidirectional neural network developed based on the well-known GRU model [47], Bi-GRU performs well in extracting global features. Compared with the LSTM unit, the GRU unit is simpler and easier to converge. A Bi-GRU unit contains two GRU units to capture forward and backward information in a sentence. In this paper, each word is inputted into two GRU units respectively. A typical GRU memory unit consists of gate structures, including reset gate and update gate. Reset gate is used to control the amount of information loss before the current time step. Update gate decides whether to neglect information from the previous time step or reserve information at the current time step. Furthermore, a series of linear and nonlinear operations are performed in the GRU unit, including dot, sum and Rectified Linear Unit (RELU). When encoding a sentence, the updating formula of a GRU unit at t time step is shown as follows:

$$r_t = \sigma\left(\boldsymbol{W}_r \cdot \left[\boldsymbol{h}_{t-1}, \hat{\boldsymbol{x}}_t\right]\right) \tag{7}$$

$$z_t = \sigma\left(\boldsymbol{W}_z \cdot \left[\boldsymbol{h}_{t-1}, \hat{\boldsymbol{x}}_t\right]\right) \tag{8}$$

$$\tilde{\boldsymbol{h}}_t = tanh\left(\boldsymbol{W}_h \cdot \left[r_t * \boldsymbol{h}_{t-1}, \hat{\boldsymbol{x}}_t\right]\right) \tag{9}$$

$$\boldsymbol{h}_t = (1 - z_t) * \boldsymbol{h}_{t-1} + z_t * \tilde{\boldsymbol{h}}_t \tag{10}$$

where t denotes the t-th time step, σ represents a sigmoid function. r_t, z_t represents the output of the reset gate and update gate respectively. Besides, $\boldsymbol{W}$ denotes a weight matrix, whose subscripts determine the category of the matrix. For example, the weight matrix of reset gate structure corresponds to $\boldsymbol{W}_r$ and update gate structure corresponds to $\boldsymbol{W}_z$ while $\tilde{\boldsymbol{h}}_t, \boldsymbol{h}_t$ denotes the original hidden state and the updated hidden state at the t time step respectively.

The input of the sentence encoding part is the combination of local feature vectors of words and pre-trained word embeddings. As shown in Fig. 1, each GRU unit receives pre-trained word embedding and local feature vectors of words in the sentence. The input of the corresponding GRU unit $\hat{\boldsymbol{x}}_t$ is shown as the following:

$$\hat{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \boldsymbol{cw}_t] \tag{11}$$

where $\boldsymbol{x}_t$ represents the pre-trained word embedding of the t-th input word and $\boldsymbol{cw}_t$ represents the local feature vector corresponding to the same word. For the sake of description, we divide Bi-GRU into forwarding GRU and backward GRU according to the direction of information transfer. Then the output of forwarding GRU at the $t-1$-th time step is denoted as $\vec{\boldsymbol{h}}_{t-1}$, and the output of the backward GRU at the $t-1$-th time step is denoted as $\overleftarrow{\boldsymbol{h}}_{t-1}$. Hence, the output of the Bi-GRU unit at t time step $\boldsymbol{h}_t$ is shown below:

$$\vec{\boldsymbol{h}}_t = GRU\left(\vec{\boldsymbol{h}}_{t-1}, \hat{\boldsymbol{x}}_t\right) \tag{12}$$

$$\overleftarrow{\boldsymbol{h}}_t = GRU\left(\overleftarrow{\boldsymbol{h}}_{t-1}, \hat{\boldsymbol{x}}_t\right) \tag{13}$$

$$\boldsymbol{h}_t = \left[\vec{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t\right] \tag{14}$$

where $\vec{\boldsymbol{h}}_t$ represents the output of the forwarding GRU at the t-th time step, and $\overleftarrow{\boldsymbol{h}}_t$ represents the output of the backward GRU at the t-th time step.

C. SENTENCE SIMILARITY CALCULATING

To capture the overall semantic information of a sentence, we concatenate the final output of forwarding GRU and backward GRU as the sentence representation. For example, for the sentence S^b, sentence representation is shown as following:

$$\boldsymbol{h}^b = \overleftarrow{\boldsymbol{h}}_0 \oplus \vec{\boldsymbol{h}}_n \tag{15}$$

here $\vec{\boldsymbol{h}}_n$ is the output of forwarding GRU at n time step, and $\overleftarrow{\boldsymbol{h}}_0$ is the output of backward GRU at 0 time step.

Then the Manhattan distance formula is adopted to measure the semantic relationship between two sentence representations. As shown in Fig. 1, the output of the model is defined as the following:

$$\mathrm{y} = exp\left(-\left\|\boldsymbol{h}^a - \boldsymbol{h}^b\right\|_1\right) \tag{16}$$

where $\boldsymbol{h}^a$ and $\boldsymbol{h}^b$ are the sentence representations respectively after the process of encoding sentences in our model.

D. TRAINING DETAILS

In this paper, we set the number of convolution filters to be 16 and the moving strip of the convolution filter is set to 1. To prevent over-fitting, the dropout rate in our model is 0.2. We also set the output dimension of the hidden layer to 50 and the value of training epochs in the model is 100.

The performance of GRU units depends on the initialization of internal parameters [48]. These parameters could get better optimization after transfer learning. Therefore, we first initialize GRU parameters with a random Gaussian function, then train our model with the dataset in the SemEval2013 task [49]. We keep the parameters of the GRU fixed in other

tasks. Besides, we adopt mean square error (MSE) as our loss function. It is shown as follows:

$$\mathcal{L}_{mse} = \sum_{i=1}^{N} \left(y_i - \acute{y}_i\right)^2 \tag{17}$$

where y_i denotes the output of our model with an input of sentence pair and $\acute{y}_i$ denotes the true similarity score of the same sentence pair in the training dataset. N represents the number of sentence pairs in the training dataset.

IV. EXPERIMENT

We conduct experiments to demonstrate the effectiveness of our proposed HM-LGSN model. In this section, we first introduce our experimental settings, then we show experimental results and make a detailed analysis.

A. DATASETS

We conduct our experiments on three different datasets which are derived from the recent SemEval competitions. The datasets contain different types of sentences with semantic relatedness:

- **Sentence Involving Compositional Knowledge** (SICK) comes from SemEval-2014 Task 1 with 10k an-notated sentence pairs. Each pair of sentences is annotated with a semantic correlation score from 1 to 5.
- **Microsoft Research Video Caption**(MSRVID) comes from SemEval-2012 Task 6 with 15k annotated sentence pairs. This task is related to movie descriptions. Each pair of sentences is annotated with a semantic correlation score from 1 to 5.
- **Semantic Textual Similarity Benchmark** (STS-B) is a collection of sentence pairs that comes from SemEval-2017 Task 1 and their contents were generated from news headlines and other sources. Each sentence pair is annotated with a semantic correlation score from 1 to 5.

In the pre-processing step, we firstly use the open-source FastText[3] word embeddings as pre-trained word embeddings. Then we use the NLTK[4] tool to preprocess sentences for text segmentation and word stemming. Finally, we adopt Min-Max Normalization to map all similarity scores into [0:1].

B. EVALUATION METRICS

We select three different evaluation metrics. Pearson correlation coefficient (γ) is widely used to measure the degree of linear correlation between two variables and it is a general evaluation metric for measuring sentence similarity. Spearman's rank correlation coefficient (ρ) is another commonly used evaluation metric that is used to evaluate the dependency between two variables. Mean square error (MSE) is a measurement reflecting the degree of difference between real value and estimated value, and it is also used as an evaluation metric of the final error in sentence similarity tasks.

[3]https://fasttext.cc/

[4] http://www.nltk.org/

C. BASELINE METHODS

We employ the following sentence similarity measuring models as baselines:

- **Skip-though [37]**: It learns sentence representations combining with an encoder-decoder architecture. Skip-though is similar to the traditional doc2vec training method [50] as it uses the current sentence to predict the context sentences in an unsupervised way.
- **Tree-LSTM [51]**: It is a variant of LSTM. Compared with the traditional LSTM, it can extract and utilize the grammatical information in sentences with a combination of the syntax tree.
- **Convet [14]**: It uses CNN to extract sentence features at different levels including word level and feature level, and proposes a new feature selection algorithm to form sentence representations.
- **MALSTM [15]**: It combines LSTM structure with Siamese network architecture. By using LSTM to extract global features, final sentence representation can be produced from the features. And it also proposes a new optimization function based on Manhattan distance to measure sentence similarity.
- **PWIM [4]**: It measures semantic similarity by constructing an interaction matrix and capturing interaction relationships between sentences. The interaction matrix is weighted according to the relationships between words and CNN is used to extract features from the matrix to capture interaction relationships.
- **Infersent [16]**: It learns general sentence representations by encoding a sentence with different neural networks, then selects representative features as sentence representations. A major innovation is that this model is trained on SNLI datasets to learn general sentence representations.
- **Gensen [38]**: It learns general sentence representations by using the multi-task learning framework. The framework combines the inductive biases of diverse training objectives to make multiple training possible.
- **CNN_LSTM [19]**: It uses CNN to extract local features and then combines LSTM to generate final sentence representations. It is worth noting that this baseline model is also based on the Siamese network.

D. EXPERIMENTAL RESULTS AND ANALYSIS

Our results on the SICK dataset are summarized in Table 2. The best result for each evaluation metric is highlighted in bold. We observe that the overall performance of our proposed model is higher, as compared to the baseline models. Since **Convet** only uses CNN to extract local features in sentences, our model achieves better results. It suggests that global features are effective in learning sentence representation. Though **Convet** extracts local features from the different granularity of the semantic unit, it ignores the effect of global features on sentence semantics. Furthermore, CNN only extracts local features in a sentence. But Bi-GRU used in

TABLE 1. The statistics and examples of datasets.

NAME	N	Sentence 1	Sentence 2	Label (similarity score)
SICK	10k	"Two young women are sparring in a kickboxing fight"	"Two women are sparring in a kickboxing match"	4.9
MSRVID	15k	"Three kids are sitting in the leaves"	"Three kids are jumping in the leaves"	3.8
STS-B	8.7K	"Someone is boiling okra in a pot"	"The man is not playing the drums"	1

TABLE 2. Performance on SICK dataset.

MODEL	γ	ρ	MSE
Skip-though	0.8655	0.7995	0.2561
Tree-LSTM	0.8676	0.8083	0.2532
Convet	0.8686	0.8047	0.2606
PWIM	0.8784	0.8199	0.2329
MALSTM	0.8822	0.8345	0.2286
Infersent	0.8850	0.8347	0.2276
Gensen	0.8880	0.8351	0.2264
CNN_LSTM	0.8876	0.8350	0.2268
HM-LGSN	**0.9018**	**0.8537**	**0.2093**

our model can extract global features that reflect the temporal relationship between words in a sentence.

Compared with **MALSTM**, our model also achieves improvement when measured by evaluation metrics γ and ρ. We can learn from that integrating local features into generating sentence representations is feasible. On the one hand, the addition of local features can alleviate the problem of insufficient feature extraction through GRU or LSTM unit structure. On the other hand, local features can reflect the relationships of adjacent words which play an important role in sentence semantics. Thus, this approach combining local features can make sentence semantic richer. Besides, the better performance of our model shows that the assumption proposed in this paper is indeed effective. By using G-CNN to extract local features of sentences as one part of word semantic features and Bi-GRU to extract global features with a combination of local features and pre-trained word embeddings, a good sentence representation can be produced.

Our model outperforms **PWIM** by 1.5% when measured in γ and 2.4% when measured in ρ. Though both **PWIM** and our model combine the variants of RNN with CNN, the results still have some difference between PWIM and our model due to the architecture of the models and the differences in CNN architecture. On the one hand, our model combines Siamese network architecture which considers the sentence as a whole and takes full account of its semantics, while **PWIM** uses interaction architecture which only considers the relationships of words or phrases between sentences and ignores the internal semantics of the sentence. On the other hand, our model uses G-CNN which is different from traditional CNN in **PWIM** to extract local features as one part of word semantic features. Besides, **PWIM** is much more complicated than our model, thus it requires more training data to achieve saturation. Compared with **Infersent** and **Gensen t**hat performed well in the GLUE[5] benchmark, our model still has a great improvement. Our model is based on the Siamese network, which is similar to the above two models. But there are some differences in the process of encoding sentences. When we start to encode sentences, we consider the local features of the sentences from different scales by using G-CNN. This approach allows for richer semantic information to be taken into account in sentence modeling to learn more comprehensive sentence representations. The selection of Bi-GRU also greatly satisfies the extraction of global features.

Compared with **CNN_LSTM**, our model achieves more satisfactory scores under different evaluation metrics. There are two main reasons for the improvements. The first reason is that when extracting local features, **CNN_LSTM** use only one type of convolution filters, so that local information of a sentence cannot be fully exploited. Instead, our model extracts local features with the help of different types of convolution filters. We select the most effective features from all local features extracted so that the features representing word semantics will be more effective. The use of bidirectional GRU in our model is the second reason for making our model more effective, it can capture forward and backward sequence information in a sentence. This bidirectional sequence information reflects the context relationship of sentences more accurately. Furthermore, GRU is simpler than LSTM, which makes our model converge with fewer data.

Table 3 reports the results on the MSRVID dataset. We find that **Gensen** performs not as well as **MALSTM** and **Infersent**, which is inconsistent with their performance in the SICK dataset. The main reason is that the contents of the data in each dataset are different. For example, the MSRVID dataset contains thousands of sentence pairs about movie reviews while the SICK dataset is made up of sentence pairs whose contents are about video description and the caption of images from Flickr.[6] As a method using multi-task learning, **Gensen** is trained with different datasets. However,

[5]https://gluebenchmark.com/
[6]https://www.flickr.com/

TABLE 3. Performance on MSRVID dataset.

MODEL	γ
Skip-though	0.9045
Tree-LSTM	0.9090
Convet	0.9101
PWIM	0.9112
MALSTM	0.9247
Infersent	0.9279
Gensen	0.9232
CNN_LSTM	0.9288
HM-LGSN	**0.9412**

TABLE 4. Performance on STS-B dataset.

STS-B	3_{rd}	2_{nd}	1_{st}	**HM-LGSN**
news	0.8113	0.8400	0.8518	**0.8762**
caption	0.8156	0.8162	0.8181	**0.8331**
forum	0.8105	0.8222	**0.8387**	0.8325

[a]We show results of the top three participating systems at the competition in Pearson correlation coefficient(γ). 1_{st}, 2_{nd}, 3_{rd} represent a first-ranked system, second-ranked system, and third-ranked system respectively in the STS-B competition.

the sentence representation extracted in this way may not be universal, which makes the performance in the MSRVID dataset decreased. We also find that our model receives the best results comparing with various baseline models including CNN_LSTM. This proves that even if the contents of datasets are different, adding local features into the sentence encoding process is very effective.

Table 4 shows the performance of our model on the STS-B dataset. Compared with the results obtained by the top three teams in the Semeval2017-Task1, we observe that our model performs better on datasets of news and caption. The team that gets the best performance in the competition proposes a model using a mixture of features. It combines sentence pair matching features, such as sequence features, syntactic features, alignment features, and n-gram overlap features, etc. with single sentence features. Then it uses three types of algorithms to get one part of similarity scores from all these mixing features. After that, they get another part of similarity scores based on a deep learning architecture. Finally, the last sentence similarity is the combination of two types of scores. It is obvious that the semantic meaning of a sentence can be indeed enriched in this way. However, comparing with our model, the model proposed by the first-rank team and second-rank team in the competition cannot extract general features from different types of features to form final sentence representation, though a lot of features in a sentence or between sentences are considered. Instead, we combine local features with word distributed features and extract global features for generating sentence representations. Our model can capture local information in sentences and global sequence relational information. Therefore, the semantic information which is contained in extracted features is richer and the generated sentence representations are also better.

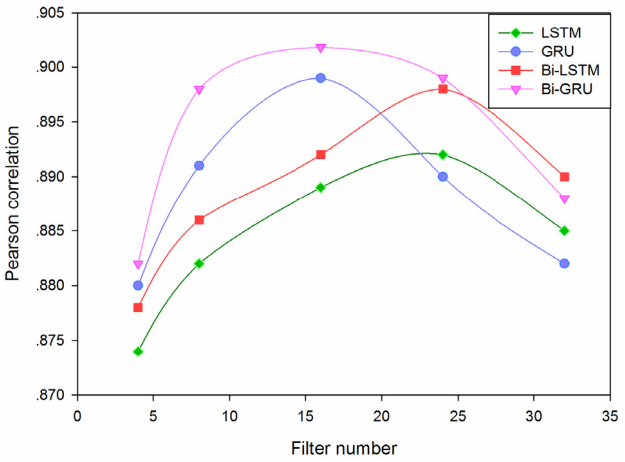

FIGURE 3. The results of tuning filter number.

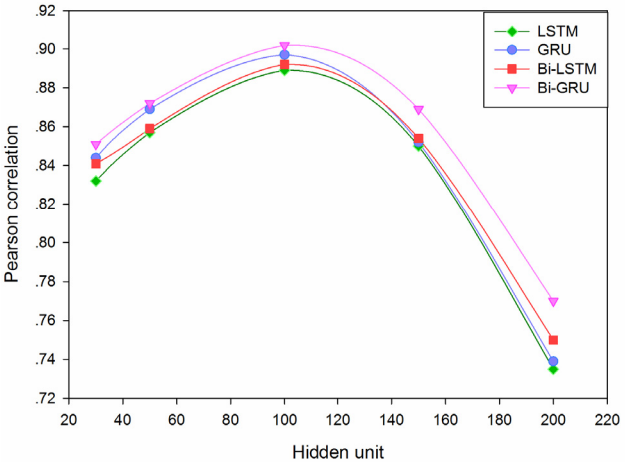

FIGURE 4. The results of tuning hidden unit.

E. PARAMETER INFLUENCE

We compare the number of convolution filters and the number of hidden units used with four different sentence encoding structures in the SICK dataset. The results are shown in Fig. 3 and Fig. 4.

As can be seen from Fig. 3, when selecting LSTM or Bi-LSTM as a sentence encoding structure, the Pearson correlation increases first and then decreases with the increasing number of convolution filters. The above two sentence encoding structures achieve the best performance when the filter number reaches 24. Instead, GRU and Bi-GRU achieve the best performance when the filter number reaches 16. The reason is that GRU is much simpler than LSTM, the same as Bi-GRU to Bi-LSTM. The former structure has fewer parameters, so it is easier to converge with less filter number. Besides, the overall trend of Pearson correlation with four different sentence encoding structures shows a lot

of similarities. We can learn from that the number of local features will affect the generation of sentence representations, with little correlation to sentence encoding structures.

As can be seen from Fig. 4, four types of sentence encoding structures all have a consistent performance with the increase of hidden units. We can learn from that the influence of the hidden units on our model performance is rather smaller than that of the choice of sentence encoding structures. When the hidden unit reaches 200, our model achieves the lowest performance. The reason is that too many hidden units make it difficult to extract abstract features, hence the resulting sentence representation is unreliable.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a hybrid model based on a Siamese network integrating G-CNN and Bi-GRU to learn sentence representations and subsequently use them for sentence similarity calculation. Our model considers local features of words in sentences and global features of sentences to generate high-quality sentence representations for measuring sentence similarity. To evaluate the performance of this proposed model, we conduct experiments on SICK, MSRVID, and STS-B datasets. Compared with the state-of-the-art baseline models, our model achieves better performance in terms of Pearson correlation coefficient, Spearman's rank correlation coefficient, and Mean-square error. Our future work will focus on learning general sentence representations by multi-task learning, which is highlighted in recent years.

REFERENCES

[1] H. Ruan, Y. Li, Q. Wang, and Y. Liu, "A research on sentence similarity for question answering system based on multi-feature fusion," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Omaha, NE, USA, Oct. 2016, pp. 507–510.

[2] B. Özateş, A. Özgür, and D. Radev, "Sentence similarity based on dependency tree kernels for multi-document summarization," in *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC)*, Portorož, Slovenia, 2016, pp. 2833–2838.

[3] G. Yasui, Y. Tsuruoka, and M. Nagata, "Using semantic similarity as reward for reinforcement learning in sentence generation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Student Res. Workshop*, Florence, Italy, 2019, pp. 400–406.

[4] 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.

[5] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," 1997, *arXiv:cmp-lg/9709008*. [Online]. Available: https://arxiv.org/abs/cmp-lg/9709008

[6] D. L. Rohde, L. M. Gonnerman, and D. C. Plaut, "An improved model of semantic similarity based on lexical co-occurrence," *Commun. ACM*, vol. 8, nos. 627–633, pp. 116–149, 2006.

[7] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[9] 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.

[10] 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.

[11] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Inf. Sci.*, vol. 373, pp. 388–403, Dec. 2016.

[12] Q. Chen, Q. Hu, J. X. Huang, and L. He, "CA-RNN: Using context-aligned recurrent neural networks for modeling sentence similarity," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 265–273.

[13] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng, "Match-SRNN: Modeling the recursive matching structure with spatial RNN," 2016, *arXiv:1604.04378*. [Online]. Available: http://arxiv.org/abs/1604.04378

[14] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1576–1586.

[15] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2786–2792.

[16] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," 2017, *arXiv:1705.02364*. [Online]. Available: http://arxiv.org/abs/1705.02364

[17] Y. Nie and M. Bansal, "Shortcut-stacked sentence encoders for multi-domain inference," 2017, *arXiv:1708.02312*. [Online]. Available: http://arxiv.org/abs/1708.02312

[18] Q. Chen, Z.-H. Ling, and X. Zhu, "Enhancing sentence embedding with generalized pooling," 2018, *arXiv:1806.09828*. [Online]. Available: http://arxiv.org/abs/1806.09828

[19] E. L. Pontes, S. Huet, A. C. Linhares, and J.-M. Torres-Moreno, "Predicting the semantic textual similarity with siamese CNN and LSTM," 2018, *arXiv:1810.10641*. [Online]. Available: http://arxiv.org/abs/1810.10641

[20] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. 6, pp. 1137–1155, 2003.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[22] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.

[23] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.

[24] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882

[25] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*. [Online]. Available: http://arxiv.org/abs/1404.2188

[26] M. Nicosia and A. Moschitti, "Accurate sentence matching with hybrid siamese networks," in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, Singapore, 2017, pp. 2235–2238.

[27] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," 2019, *arXiv:1908.10084*. [Online]. Available: http://arxiv.org/abs/1908.10084

[28] Z. Jingling, Z. Huiyun, and C. Baojiang, "Sentence similarity based on semantic vector model," in *Proc. 9th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput.*, Guangdong, China, Nov. 2014, pp. 499–503.

[29] 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, 2018, pp. 3890–3902.

[30] N. Madnani, J. Tetreault, and M. Chodorow, "Re-examining machine translation metrics for paraphrase identification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Montréal, QC, Canada, 2012, pp. 182–190.

[31] S. Fernando and M. Stevenson, "A semantic similarity approach to paraphrase detection," in *Proc. 11th Annu. Res. Colloq. UK Special Interest Group Comput. Linguistics*, 2008, pp. 45–52.

[32] D. Das and N. A. Smith, "Paraphrase identification as probabilistic quasi-synchronous recognition," in *Proc. Joint Conf. 47th Annu. Meeting ACL, 4th Int. Joint Conf. Natural Lang. Process. (AFNLP)*, Singapore, vol. 1, 2009, pp. 468–476.

[33] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 2793–2799.

[34] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 670–680.
[35] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 2835–2841.
[36] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, Beijing, China, vol. 1, 2015, pp. 1556–1566.
[37] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Proc. 28th Int. Nat. Conf. Neural Inf. Process. Syst.*, Montréal, QC, Canada, 2015, pp. 3294–3302.
[38] S. Subramanian, A. Trischler, Y. Bengio, and C. J Pal, "Learning general purpose distributed sentence representations via large scale multi-task learning," 2018, *arXiv:1804.00079*. [Online]. Available: http://arxiv.org/abs/1804.00079
[39] A. Chaturvedi, O. Pandit, and U. Garain, "CNN for text-based multiple choice question answering," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Melbourne, VIC, Australia, 2018, pp. 272–277.
[40] S. Wang, M. Huang, and Z. Deng, "Densely connected CNN with multi-scale feature attention for text classification," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Stockholm, Sweden, 2018, pp. 4468–4474.
[41] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Santiago, Chile, 2015, pp. 373–382.
[42] J. Shin, Y. Kim, S. Yoon, and K. Jung, "Contextual-CNN: A novel architecture capturing unified meaning for sentence classification," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Shanghai, China, Jan. 2018, pp. 491–494.
[43] W. Yin and Y. Pei, "Optimizing sentence modeling and selection for document summarization," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 1383–1389.
[44] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.
[45] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078
[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[47] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555
[48] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. ICML Workshop Unsupervised Transf. Learn.*, Bellevue, WA, USA, 2012, pp. 17–36.
[49] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "SEM 2013 shared task: Semantic textual similarity," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics (SEM)*, Atlanta, GA, USA, vol. 1, 2013, pp. 32–43.
[50] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, Bejing, China, 2014, pp. 1188–1196.
[51] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," 2015, *arXiv:1503.00075*. [Online]. Available: http://arxiv.org/abs/1503.00075

YULONG LI received the B.Sc. degree from the Hunan University of Science and Technology, China, in 2017, where he is currently pursuing the M.Sc. degree in software engineering with the School of Computer Science and Engineering. His research interests include nature language processing and information retrieval.

DONG ZHOU received the Ph.D. degree from the University of Nottingham, U.K., in 2009. He worked as a Research Fellow at the Centre for Next Generation Localization, Trinity College Dublin, Ireland, from 2008 to 2012. He is currently a Professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, China. His current research interests include information retrieval, natural language processing, machine learning, and data mining.

WENYU ZHAO received the M.Sc. degree in software engineering from the Hunan University of Science and Technology, China, in 2018, where she is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. Her research interests include information retrieval, natural language processing, and personalized search.