IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 15, 2018, accepted December 8, 2018, date of publication December 18, 2018,
date of current version January 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2888508

Multi-Gram CNN-Based Self-Attention Model for Relation Classification

CHUNYUN ZHANG[1,2], CHAORAN CUI[1], SHENG GAO[3], XIUSHAN NIE[1], WEIRAN XU[3], LU YANG[1], XIAOMING XI[1], AND YILONG YIN[2]

[1]School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China
[2]Shandong University, Jinan, China
[3]Beijing University of Posts and Telecommunications, Beijing, China

Corresponding authors: Chunyun Zhang (zhangchunyun1009@126.com) and Yilong Yin (ylyin@sdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703234, Grant 61701281, Grant 61671274, Grant 61703235, Grant 61573219, and Grant 61701280, in part by the Postdoctoral Science Foundation of China under Grant 2018M632674, in part by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions, Higher Educational Science and Technology Program of Shandong Province, under Grant J17KA065, in part by the Natural Science Foundation of Shandong Province under Grant ZR2016FQ18 and Grant ZR2017QF009, in part by the Beijing Nova Program Grant under Grant Z171100001117049, and in part by the Beijing Natural Science Foundation under Grant 4162044.

ABSTRACT Relation classification is a crucial ingredient in numerous information-extraction systems and has attracted a great deal of attention in recent years. Traditional approaches largely rely on feature engineering and suffer from the limitations of the domain adaption and the error propagation. To overcome the above-mentioned problems, many deep neural network-based methods have been proposed; however, these methods cannot effectively locate and utilize the relation trigger features. To locate the relation trigger features and make full use of them, we propose a novel multi-gram convolution neural network-based self-attention model with a recurrent neural network framework. The multi-gram conventional neural network attention model can learn the adaptive relational semantics of inputs based on the fact that a relation can be totally defined by the shortest dependency path between its two entities. With the learned relational semantics, we can obtain the corresponding importance distribution over input sentences and locate the relation trigger features. For effective information propagation and integration, we utilize a bidirectional gated recurrent unit to encode the high-level features during recurrent propagation. The experimental results on two benchmark datasets demonstrate that the proposed model outperforms most of the state-of-the-art models.

INDEX TERMS Erelation extraction, multi-gram, attention, CNN, RNN.

I. INTRODUCTION

Relation extraction is a crucial component in the field of natural language processing (NLP) and plays an essential role in various scenarios such as information extraction [1], question answering [2], ontology learning [3], etc. In recent decades, many related tasks, e.g., message understanding conference (MUC) [4], automatic content extraction (ACE) [5], and knowledge base population (KBP) [6] of the text analysis conference (TAC) [7], arose and facilitated the development of relation extraction technology. Generally speaking, one of the most frequently used paradigms for relation extraction is to construct a relation classifier to perform relation classification especially for limited relation types.

Relation classification is the task of identifying the semantic relation between two nominal entities from several relation types. For instance, given an example input:

"*The* $[treaty]_{e1}$ *established a double majority* $[rule]_{e2}$ *for council decisions.*"

with annotated target entity mentions e1="*treaty*" and e2="*rule*". The goal is to automatically recognize that this sentence expresses a *Cause–Effect* relationship between e1 and e2, for which we use the notation *Cause–Effect (e1, e2)*. Accurate relation classification can significantly facilitate sentence interpretations, discourse processing and higher-level NLP tasks. Therefore, relation classification has attracted considerable attention from researchers [8]–[10].

However, reliable relation classification remains an open problem. One of the major challenges is how to learn the relation indication information. It is obvious that there are different ways to express the same type of relationship. In other words, the challenging variability can be lexical, syntactic, or even pragmatic in nature. As mentioned in [8],

a relation is mainly defined by the shortest dependency path of the target two named entities (entity pair). Hence, an effective solution should be able to account for learning useful semantic and syntactic features between the two target entities, which is the true expression of the target relationship.

Traditional relation classification approaches rely largely on feature representation, or kernel design. The former usually incorporates a set of features derived from the output of an explicit linguistic preprocessing step [1], [11]–[13]. It is difficult to improve the model performance if the feature set is not properly chosen. The latter depends largely on the designed kernel, which summarizes all data information. The fundamental issue of designing an effective kernel becomes crucial, and includes convolution tree kernels [14], subsequence kernels [15], and dependency tree kernels [8]. Although these methods benefit from well-established NLP tools [8], [9], [16], they suffer from the limitations of domain adaption and error propagation. This is because they cannot learn robustness underlying features automatically.

Recently, deep neural architectures have attracted increased attention. Convolution neural networks (CNNs), recurrent neural network (RNNs), and their variations [17]–[20] are the mainstream network architectures. These architectures are capable of learning relevant representations and features without extensive manual feature engineering or using external resources. They have achieved state-of-the-art performance in many tasks of NLP, including machine translation, relation classification, and sentiment analysis. For relation classification, the most representative progress has been made by Zeng *et al.* [10]. They proposed a CNN-based approach that obtains quite competitive results without any extra knowledge resource and NLP modules. Following the success of CNN, there are some valuable models such as multi-window CNN [21], CR-CNN [22], and NS-depLCNN [23]. A potential problem of CNN is that such a model can learn only local patterns. In particular, simply increasing the window size of the convolutional filters does not work because this will lose the strength of CNNs in modeling local or short distance patterns. To tackle this problem, Nguyen and Grishman [21] proposed a CNN model with multiple window sizes for filters, which allows patterns of different lengths to be learned. Although this method is promising, it is still inadequate in learning long-distance patterns.

To overcome the drawback of CNNs, researchers tend to use RNN models to learn long distance semantic information. A RNN is an extension of a conventional feedforward neural network, which is able to handle a variable-length sequence input. The RNN model was first introduced in relation classification by Socher *et al.* [17]. They proposed a recursive neural network model to learn compositional vector representations for phrases and sentences of arbitrary syntactic type and length. At the same time, Liu *et al.* [24] proposed a DepNN model, which uses a recursive neural network to model the subtrees, and a CNN to capture the most important features on the shortest path. Subsequently, some variations of RNNs, long short term memory (LSTM) [25] and gated recurrent unit (GRU) [26], and some more elaborate variants have been proposed, including bidirectional LSTMs [19] and bidirectional tree-structured LSTM-RNNs [27]. Several recent works also reintroduced a dependency tree-based design, e.g., RNNs operating on syntactic trees [28] and the SDP-LSTM model [25]. Although these recent models can achieve solid results, they often fail to identify critical cues of relation semantics and cannot make full use of this critical information.

To alleviate the aforementioned problems, an attention model is introduced into the NLP area, which allows models to learn the importance distribution over the inputs. In the field of NLP, attention models have been successfully applied to sequence-to-sequence learning tasks, such as machine translation [29], abstractive sentence summarization [30], and question answering [31]. These models have generally been used to facilitate alignment of the input and output. However, relation classification is a sequence-label task, and how to obtain the relation attention vector is the pivotal issue.

To the best of the authors' knowledge, there are several related works utilizing the attention mechanism in their relation classification systems [32]–[35] in the last three years. These attention mechanisms are mainly used to extract word-level and sentence-level importance of the input data, and the sentence-level attention mechanism is mostly used in distance supervised relation classification systems [36]–[38]. The word-level attention mechanism can be summarized into two categories. The first category is random attention mechanism, which randomly initializes the attention vector, and learns it using a general end-to-end deep neural network framework without any relation indication information. The other category is relation indicate (relation supervised) attention mechanism, which makes use of the semantic relation information contained in the input sentence to automatically learn the corresponding attention vector. The former does not make use of relation indication information and tends to yield over-fitting problem. The latter can make use of relation indication information. However, it only makes use of simple features, such as entity-pair feature. Owing to the limited scale of the training dataset, the obtained attention vector of both methods may be suboptimal for the heterogeneous structure of relation expressions.

To overcome the disadvantages of these two attention categories, we present a novel adaptive attention model in a general deep neural network framework, which can make full use of relation indication information. Based on the fact that a relation is mostly defined by its target entity pair and the shortest dependency path between them, we propose a multi-gram CNN-based self-attention framework to extract relation indication information from the definition features mentioned above. Based on the extracted relation indication information, we can generate the relation attention vector correspondingly. With the learned relation attention vector, we can obtain the "importance" distribution over inputs by computing the similarity of each word with the relation attention vector.

For effective information propagation and integration, we utilize bidirectional GRU (BiGRU) to extract high-level features during recurrent propagation. The main contributions of this paper are as follows.

1. The proposed framework is a relation-adaptive attention model in a general RNN architecture. It can locate relation trigger features of inputs by using a self-attention model.
2. To better mine relation indication information, we proposed a novel multi-gram CNN-based self-attention model. It utilizes a multi-gram CNN framework to learn the relation indication information from the shortest dependency path of the two target entities. It is more reasonable than models that only use an entity pair as the relation indication information.
3. We evaluate our method on two commonly used relation classification datasets. Experimental results show that our method is effective and outperforms several competitive baseline methods.

This paper is an improvement and extension of our previous work [39]. The rest of this paper is organized as follows. In Section II, we give a brief overview of existing relation classification models. In Section III, we present our modeling framework including the input embedding layer, the BiGRU layer, the multi-gram CNN-based attention layer, and the output layer. Section IV presents the settings of each layer of the deep neural framework, and the experimental results on the classic relation classification datasets. We draw our conclusions in Section V.

II. RELATED WORK

Relation classification is a widely studied task in the NLP community. There are various relation classification methods, and they can be categorized into three classes: feature-based [11], [12], [40], kernel-based [8], [41], [42], and neural-network-based [10], [19], [22], [25].

In feature-based approaches, different sets of features are extracted and fed to a chosen classifier (e.g., support vector machine (SVM)). In general, three types of features are often used: lexical features, syntactic features, and semantic features. Lexical features concentrate on the entities of interest, e.g., entities per se, entity part-of-speech (POS), entity neighboring information. Syntactic features include chunking, parse trees, etc. Semantic features are exemplified by the concept hierarchy, entity class, entity mention. Kambhatla [11] used a maximum entropy model to combine these features for relation classification. However, different sets of handcrafted features are largely complementary to each other (e.g., hypernyms versus named-entity tags), and thus it is hard to improve performance in this way [40].

Kernel-based approaches specify some measure of similarity between two data samples without explicit feature representation. Zelenko *et al.* [41] calculated the similarity of two trees by using their common subtrees. Bunescu and Mooney [8] proposed a shortest path dependency kernel for relation classification. Its main idea is that a relation is defined by the dependency path between two given entities. Wang [42] presented a systematic analysis of several kernels and showed that relation extraction can benefit from combining convolution kernel and syntactic features. Plank and Moschitti [43] introduced semantic information into kernel methods in addition to considering structural information only. However, one potential difficulty of kernel methods is that all data information is completely summarized by the kernel function (similarity measure), and thus designing an effective kernel becomes crucial.

In general, these methods mentioned above depend either on carefully handcrafted features, or on elaborately designed kernels. The carefully handcrafted features are often chosen on a trial-and-error basis, whereas the elaborately designed kernels are often derived from other pre-trained NLP tools or lexical and semantic resources. Although such approaches can benefit from the external NLP tools to discover the discrete structure of a sentence, syntactic parsing is error-prone and relying on its success may also impede performance [44]. Further downsides include their limited lexical generalization abilities for unseen words and their lack of robustness when applied to new domains, genres, or languages. In all, these methods cannot learn robustness underlying features automatically.

Deep neural networks have emerged recently and can learn robust underlying features automatically with promising results. Zeng *et al.* [10] used a deep CNN to extract lexical and sentence-level features. Based on CNNs, dos Santos *et al.* [22] proposed a Ranking CNN (CR-CNN) model with a class embedding matrix. Xu *et al.* [23] leveraged CNNs to learn representations from shortest dependency paths, and addressed the relation directionality by special treatments on sampling. However, even though CNN performs better on recognizing consecutive patterns for relation mentions, it is not suitable for learning long-distance semantic information. To address this issue, RNN-based models have been utilized [17], [18], [24], [45], [46]. The MV-RNN model proposed a RNN model to learn compositional vector representations for phrases and sentences of arbitrary syntactic type and length. The DepNN method utilized a RNN to model subtrees, and a CNN to capture the most important features on the shortest path. Unfortunately, standard RNNs suffer from the problem of vanishing or exploding gradients [47], [48], where gradients may grow or decay exponentially over long sequences. This makes it difficult to model long-distance correlations in a sequence. Hence, the LSTM unit has been utilized. It was proposed and extended [48]–[50] with the motivation on an analysis of recurrent neural nets [51], which found that long and time lags were inaccessible to existing architectures, because backpropagated error either blows up or decays exponentially. A LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. Each one contains one or more recurrently connected memory cells and three multiplicative units: the input, output, and forget gates that provide continuous analogs of write, read, and reset operations for the cells.

LSTM has achieved the best known results in handwriting recognition [52], speech recognition [53], and computer vision [54]–[56]. In recent years, since 2012, LSTM has achieved great success in the NLP domain [57]–[60]. Based on the good performance of LSTM, some elaborate variants have been proposed, including GRU and bidirectional LSTM (BLSTM). GRU was first proposed by Cho *et al.* [61] to make each recurrent unit adaptively capture dependencies of different time scales. Compared with LSTM, the GRU framework can achieve considerable performance improvement but with a lower computational cost. BLSTM [19] is another variant of LSTM, which adopts a bidirectional LSTM to identify relations and confirmed the superiority against the unidirectional framework. In addition to the deep neural network frameworks, several recent works also reintroduced a dependency tree-based design, e.g., RNNs operating on syntactic trees [28] and the SDP-LSTM model [25].

In parallel, the concept of "attention" has gained popularity recently in training neural network models [36], [62]–[65]. Such models iteratively process their input by selecting relevant content at every step. This basic idea significantly extends the range of applicability of end-to-end training methods, for instance, making it possible to construct networks with external memory [66]. Since Bahdanau *et al.* [29] introduced the attention mechanism in machine translation in 2014, it has attracted a lot of interest in the NLP domain, including in machine translation [29], abstractive sentence summarization [30], and question answering [31].

Unlike these sequence-to-sequence problems, relation classification is a sequence-to-label problem, the pivotal issue of which is finding the key supervisory information, which is equal to the alignment factor in sequence-to-sequence problem. Aiming at this issue, hierarchical attention networks (HN-ATT) [67] put forward a solution by randomly initializing an equal-length vector as the word-level attention vector. It computed the dot product of the attention vector and each word-level vector to generate corresponding attention weights. Inspired by this, attention-based BLSTM [33] employed the same mechanism to learn the "importance" distribution over the whole sentence. Different from the word-level attention, there is another attention paradigm that tries to learn sentence-level attention. This paradigm is used mostly in distant supervise methods [36]–[38]. Our proposed multi-gram CNN-based self-attention framework belongs to the word-level attention methods and we only discuss the word-level attention paradigm in this paper.

III. THE PROPOSED MODEL

Given a sentence s with a labeled pair of entity mentions e_1 and e_2, relation classification is the task of identifying the semantic relation holding between e_1 and e_2 among a set of candidate relation types [8]. As the shortest dependency path of the two mentioned entities concentrates on the most relevant information while diminishing less relevant noise in the relation definition [25], it is crucial to learn to extract these cues to make an accurate prediction. To this end, we proposed a novel multi-gram CNN-based attention model on a RNN. For effective information propagation and integration, our model leverages BiGRU as the general framework during recurrent propagation.

A schematic overview of our architecture is given in Figure 1. As shown in Figure 1, the proposed model mainly consists of four components.

- **Input embedding layer**: encodes each word of the input sentence into word vector representation by exploiting word-level and position information.
- **BiGRU layer**: utilizes BiGRU to obtain high-level semantic features from step 1.
- **Multi-gram CNN-based attention layer**: A multi-gram CNN framework is used to learn the relation-specific attention vector from the shortest dependency path of the mentioned entity pair. Through weighting using this vector, the outputs of BiGRU are integrated to generate a sentence-level vector.
- **Output layer**: the sentence-level feature vector is finally used for relation classification.

In the following subsections, we describe these four components in detail.

A. INPUT EMBEDDING

For the relation classification task, each input is composed of a word sequence $s = \{w_1, w_2, \ldots, w_n\}$ and its corresponding labeled entity pair (e_1, e_2). Hence, we can incorporate two embeddings into the input embedding: word embedding and position embedding.

1) WORD EMBEDDING

Word embedding (WE) transforms words into real-valued vector representations that capture syntactic and semantic information about the words. Given a WE matrix $W_v \in R^{d_w \times |V|}$, the word w_i of s can be encoded as w_i^w by looking up this matrix. Here, V is a fixed-size vocabulary, and d_w is the size of word embedding.

Note that to lead to initializing the parameters at a good point of our proposed model, we use word2vec trained word embeddings to initialize the word embedding layer. The word2vec was developed by Mikolov *et al.* [16], [68] to minimize the computational complexity of word embedding training. Ouyang *et al.* [69] showed that using word2vec trained word embeddings to initialize the input of deep neural networks can lead to initializing the parameters at a good point and efficiently improve the performance of the proposed deep models.

The word2vec includes two efficient word embedding model, namely the continuous Skip-gram model (Skip-gram) and continuous bag-of-words model (CBOW). As shown in Figure 2, there is no hidden layer existing in the two models, hence, the intensity of correlation between words is directly measured by the inner product between word embeddings. This means that they perform more effectively

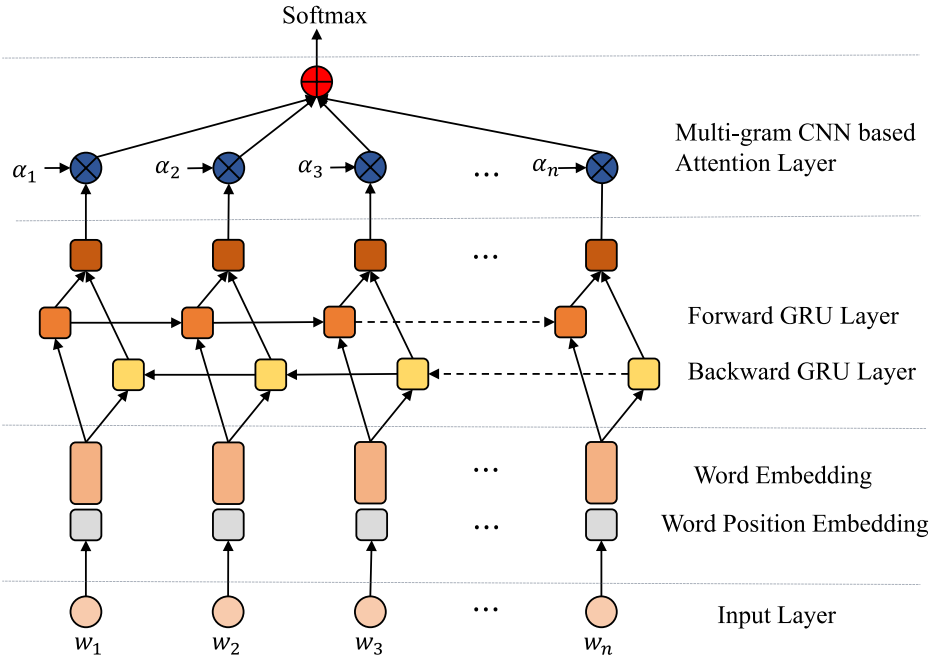

FIGURE 1. Schematic overview of our proposed model.

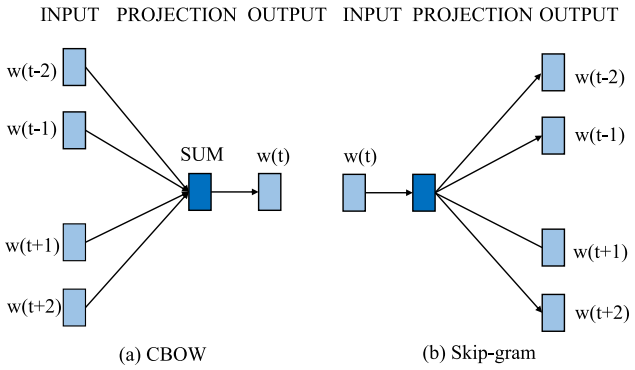

FIGURE 2. The CBOW and skip-gram architecture of word2vec [16].

on the calculation complexity. Such simpler models might not produce as precise distributed representation as deep neural networks on relatively small corpus; however, it can possibly be trained on much more data efficiently, and the quality of representation is proved to be superior.

2) WORD POSITION EMBEDDING

It has been verified that crucial information for identifying relations tends to be concentrated on the words close to the target entity pairs [10]. Hence, we incorporate word position embeddings (WPEs) to reflect the relative distances from the ith word to the two labeled entities. For the sentence shown in Section I, the relative distances of word "*established*" to entity e_1 "*treaty*" and entity e_2 "*rule*" are -1 and 4, respectively. Each relative distance is mapped to a randomly initialized position vector in R^{d_p}, where d_p is a hyper-parameter. For word w_i, there are two distance vectors $w_i^{p_1}$ and $w_i^{p_2}$ with regard to entity e_1 and entity e_2, respectively. The overall input representation for word w_i is $w_i^E = [(w_i^w)^T, (w_i^{p_1})^T, (w_i^{p_2})^T]^T$.

B. BIGRU ENCODER

GRU is a simpler variant of LSTM. It was first proposed by Cho *et al.* [61] to make each recurrent unit adaptively capture dependencies of different time scales. Compared with the LSTM, GRU shares many of the same properties and greatly reduces the number of parameters. Based on this good property, we choose GRU to model sequence data for our relation classification system.

The architecture of the GRU is illustrated in Figure 3. Typically, a GRU-based RNN has two gates: a reset gate r and an update gate z. The update gate controls how much information from the previous hidden state will carry over to

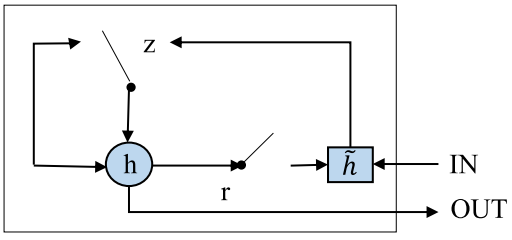

FIGURE 3. Illustration of GRU. Here r and z are the reset and update gates, and h and are the activation and the candidate activation [61].

the current hidden state. At step t, r_t is computed by

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right), \tag{1}$$

where x_t and h_{t-1} are the input in time t and the previous hidden state, σ is the logistic sigmoid function, and W_r and U_r are weight matrices that are learned.

Similarly, the update gate z is computed by

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right). \tag{2}$$

The output hidden state h_t is then computed by

$$h_t = (1 - z_t)\, h_{t-1} + z_t \widetilde{h}_t, \tag{3}$$

where $\widetilde{h}_t$ is the new hidden state. Its computation is analogous to traditional RNN:

$$\widetilde{h}_t = \tanh\left(W_h x_t + U_h\left(h_{t-1} \odot r_t\right) + b_h\right). \tag{4}$$

The difference is that the reset gate r_t is used to determine how to combine the new input with the previous memory. When r_t is close to zero, this means the previous hidden state is compelled to be ignored.

For many sequence modeling tasks, it is beneficial to have access to complete, sequential information about all future as well as past context. However, a standard GRU network only processes past context along an input sequence in temporal order. In this paper, we utilize a BiGRU architecture to model sequence data. The basic idea of BiGRU is to present each training sequence forward and backward to two separate recurrent nets, both of which are connected to the same output layer. This means that for every point in a given sequence, the network has complete, sequential information about all points before and after it. The structure of BiGRU is shown in Figure 1.

From Figure 1, we can see that the network contains two sub-networks for the forward and backward sequence, respectively. By using the BiGRU architecture, the two direction hidden states of word w_i are obtained as $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$. Based on this, the output hidden state of word w_i is represented as

$$h_i^* = \left[\overrightarrow{h_i}, \overleftarrow{h_i}\right]. \tag{5}$$

Here, we use vector concatenation to combine the forward and backward pass hidden states.

C. MULTI-GRAM CNN-BASED SELF-ATTENTION MODEL

Attentive neural networks were first proposed by Mnih *et al.* [62] in computer vision to ignore the clutter present in an image by centering the retina on the relevant regions. Based on its success, the attention model was introduced into the NLP domain by Bahdanau *et al.* [29] to solve machine translation tasks. Soon after that, the attention model enjoyed great success in sequence-to-sequence learning tasks in NLP because it can automatically calculate an ''importance'' distribution for words in a text sequence. Compared with these sequence-to-sequence tasks, relation classification tasks are a different kind of scenario, know as sequence-to-label problems. To the best of the authors' knowledge, the key technology of the attention model in the sequence-to-sequence scenario is obtaining the alignment factor. Similarly, for sequence-to-label tasks, how to obtain task-oriented (label-related) attention vectors is important. This means that it is vital to obtain the relation indication information for the relation classification task.

In this section, we design a self-attention model based on the idea of modeling attention layers with respect to relation indication information. As has been verified, relations are mostly defined by the target entity pair and the shortest dependency path between its two entities. This means that relation indication information can be learned from these features. Hence, we utilize a multi-gram CNN-based attention model to learn this relation indication information and generate the corresponding attention vector. By computing similarities between each word with the attention vector, we can obtain the ''importance'' distribution of words included in the input sentence.

1) THE SHORTEST DEPENDENCY PATH

The dependency parse tree is naturally suitable for relation classification because it focuses on the action and agents in a sentence [70]. As reported by Bunescu and Mooney [8], the shortest dependency path between two entities plays a vital role in relation classification task. Because it condenses most illuminating information for entities$\acute{a}$ relation while diminishing less relevant noise.

We take the sentence mentioned in Section I as an example: it expresses a *Cause–Effect* relationship between ***treaty*** and ***rule***. The shortest dependency path between ***treaty*** and ***rule*** is as follows:

$$\underset{e_1}{treaty} \leftarrow \underset{r_1}{nsubj} \leftarrow \underset{w_1}{established} \rightarrow \underset{r_2}{dobj} \rightarrow \underset{e_2}{rule}$$

We can see that the shortest path includes the structure of the ''*established dobj rule*,'' which helps in judging the *Cause–Effect* relation.

2) THE MULTI-GRAM CNN-BASED ATTENTION FRAMEWORK

The multi-gram CNN architecture is illustrated in Figure 2.

As shown in Figure 4, a pair of entities and words between them are encoded with the corresponding word embedding by looking up the embedding matrix W^v. By using a sliding window of size k centered around the jth word, we encode k successive words into a vector $g_j \in R^{kd_{cnn}}$ to incorporate contextual information as

$$g_j = \left[\left(w^E_{j-\frac{k-1}{2}}\right)^T, \ldots, \left(w^E_{j+\frac{k-1}{2}}\right)^T\right]^T. \tag{6}$$

Note that an extra padding token is repeated multiple times to ensure that the beginning and end of the input are well-defined.

Then we use the CNN network to learn relation indication information by

$$Q = ReLU\left(W^m \cdot G\right), \tag{7}$$

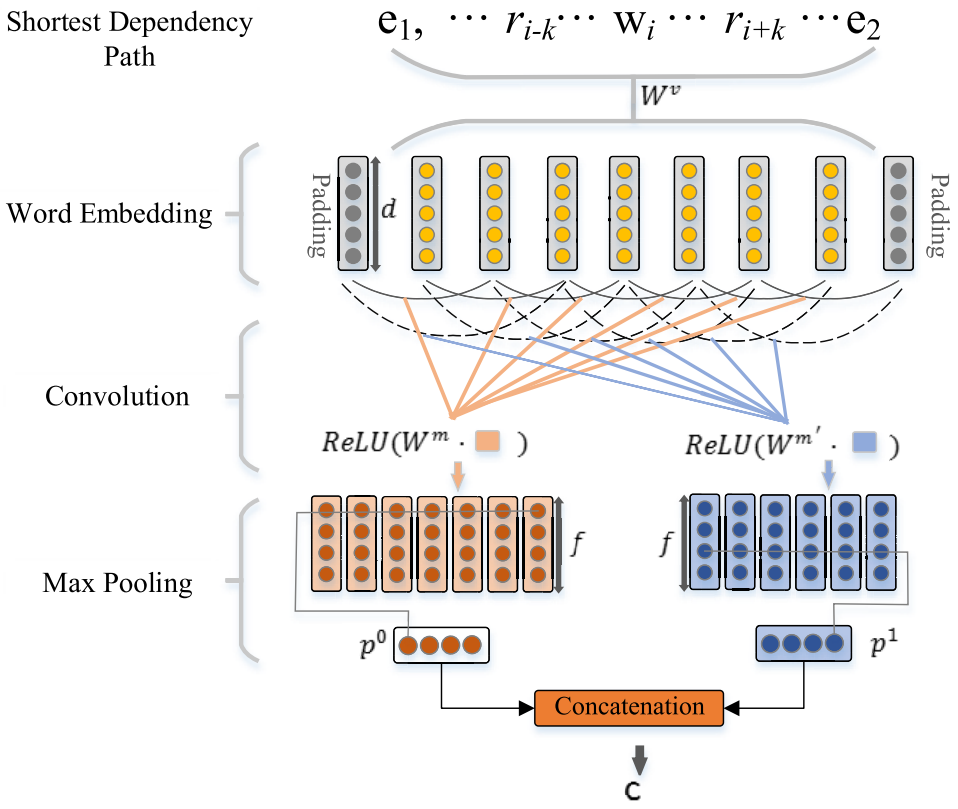

FIGURE 4. Architecture of the multi-gram CNN-based self-attention model.

where $Q \in R^{f \times N}$ and f represents the number of feature mapping layers.

A max-pooling layer is as follows:

$$p_i = \max_j \{q_{ij}\}, \quad \forall i = 1, \ldots, N. \tag{8}$$

By using the multi-gram strategy, we combine multi-gram features with a concatenation operation. The final output of the relation-specific attention is shown in the following equation:

$$c = \left(p^0, p^1, \ldots, p^K\right). \tag{9}$$

Figure 5 illustrates the strategy of utilizing tri-gram, four-gram, and five-gram features in the attention mechanism.

Next, c is fed into a fully connected layer to keep the dimension consistent with the hidden layer activation value in BiGRU (as shown in Figure 5):

$$a_s = \tanh(W_a c + b_a), \tag{10}$$

where a_s is the relation attention vector.

Let $H = \left[h_1^*, h_2^*, \ldots, h_n^*\right]$ be a matrix consisting of outputs of BiGRU, which can be seen as the word-level feature vector. Subsequently, we compute the contribution of w_i as the similarity of the relation attention vector and word-level vector. Through a softmax operation, we obtain a normalized

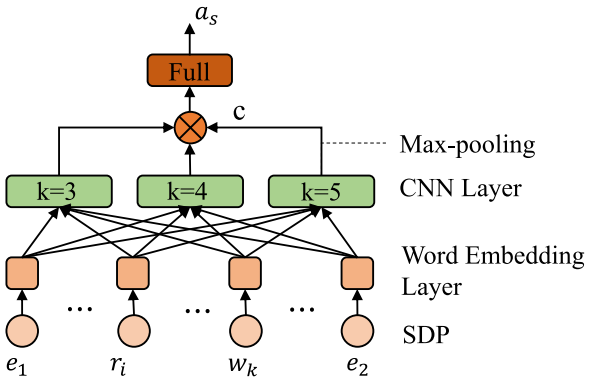

FIGURE 5. Multi-gram CNN attention mechanism based on tri-gram, four-gram, and five-gram features.

importance weight α_i

$$h_i = tanh(h_i^*), \tag{11}$$

$$\alpha_i = \frac{exp\left(h_i{}^T a_s\right)}{\sum_{i=1}^{n} exp\left(h_i{}^T a_s\right)}. \tag{12}$$

Then, we adopt two different schemes to calculate sentence representation as follows [36].

a: VECTOR SUM

BiGRU outputs are aggregated by vector sum operation weighted on the attention weight α_i, and we obtain the final

sentence representation s:

$$s = \sum_{i=1}^{n} \alpha_i h_i^*. \tag{13}$$

b: MAX-POOLING

The other scenario to produce sentence representation is the max-pooling operation. It is calculated as follows:

$$s_j = max_i \left(\alpha_i h_{ij}^* \right), \tag{14}$$

where s_j is the jth dimension of s.

D. CLASSIFICATION

After the operations mentioned above, the obtained sentence representation vector is a high-level sentence representation, which can be directly used for classification. In this setting, we use a softmax classifier to predict label y from a discrete set of classes Y for a sentence s:

$$p(y|s) = softmax \left(W_c s + b_c \right). \tag{15}$$

The relation label with highest probability value is identified as the ultimate result:

$$\hat{y} = \underset{y}{argmax}\, p(y|s). \tag{16}$$

E. REGULARIZATION

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, over-fitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with over-fitting by combining the predictions of many different large neural nets at test time.

Dropout is a technique for addressing this problem. It was proposed by Hinton *et al.* [71] in 2012. A schematic diagram of the dropout is shown in Figure 6. The key idea of dropout is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different ''thinned'' networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces over-fitting and gives major improvements over other regularization methods. In this paper, we apply a dropout strategy on the embedding layer, attention layer, and the penultimate layer.

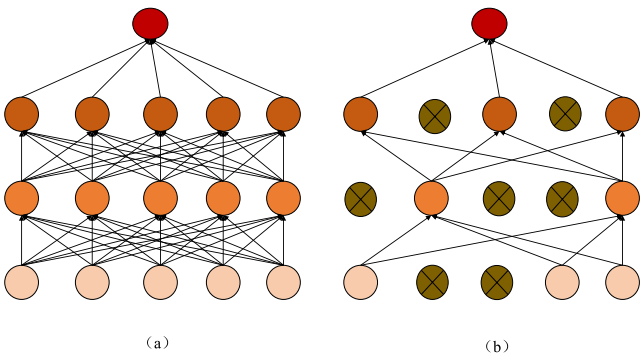

FIGURE 6. Schematic diagram of the dropout [71]. (a) Standard neural network. (b) After applying dropout.

In addition to the dropout strategy, we also use max norm constraints. The max norm constraints enforce an absolute upper bound on the magnitude of the weight vector for every neuron and use projected gradient descent to enforce the constraint. In practice, this corresponds to performing the parameter update as normal, and then enforcing the constraint by clamping the weight vector w of every neuron to satisfy $||W||_2 < \varepsilon$. Some works [72] report improvements when using this form of regularization. One of its appealing properties is that the network cannot ''explode'' even when the learning rates are set too high because the updates are always bounded.

IV. EXPERIMENTS

Our experiments are intended to demonstrate that our neural models with multi-gram CNN-based self-attention can locate key relation features and further improve the final classification performance. To this end, we introduce the experimental results and analysis on the commonly used SemEval-2010 Task 8 dataset in supervised relation classification methods. In addition, to further verify the performance of our proposed model, we apply it on a small dataset that comes from the widely used dataset in distant supervised methods.

A. DATASET AND SETTINGS

The SemEval-2010 Task 8 benchmark [73] is the one of the most commonly used datasets in supervised relation classification methods. This dataset has 10717 sentences that consist of the predefined training set of 8000 examples and a test set of 2717 examples. Within the training set, we randomly select 800 examples as the validation set. In the target corpus, each sentence has been annotated with two target entities and a unique relation label. There are nine actual relation classes, together with an artificial class *Other*. The relation types are:

- *Cause–Effect*
- *Component–Whole*
- *Content–Container*
- *Entity–Destination*
- *Entity–Origin*
- *Message–Topic*
- *Member–Collection*
- *Instrument–Agency*
- *Product–Producer*

In particular, as mentioned above, each actual relation class has a reversed version, such as *Content-Container (e1,e2)* and *Content-Container (e2,e1)*. This is to say, for a specific actual relation, the sentence with entity pair in opposite order belongs to a different relation class. However, *Other* does not

have a reversed version. Hence, the total number of relation classes is 19. The official evaluation metric is adopted to evaluate our models. The final performance is based on macro-averaged F1-score over nine actual relation classes (excluding *Other*).

We use the released word embedding set Google Newsvectors-negative300.bin to initialize our embedding layer, which is trained by Mikolov's word2vec tool.[1] In this paper, we set word embeddings to be 300-dimensional and POS embeddings to be 10-dimensional.

Beyond that, other matrices in our model are initialized randomly following a Gaussian distribution. We utilize AdaDelta [74] with a mini-batch size to learn network parameters. The optimal hyper-parameters are determined by a cross-validation procedure. For the max norm constraints parameter ε, it is typically chosen as 3 or 4; we set it as 3 in our proposed model. Detailed settings are presented in Table 1.

TABLE 1. Hyper-parameter settings.

Parameter	Parameter Name	Value
d_w	word and position	300
d_p	embedding size	10
d_h	embedding size of GRU	100
d_{cnn}	embedding size of CNN	300
ρ_w	dropout size of word	0.7
ρ_a	embedding layer, attention layer	0.6
ρ_m	and penultimate layer	0.2
k	window size of CNN	3, 4, 5
λ	learning rate	1.0
n_b	batch size	20
ε	max norm constraints parameter	3

B. IMPACT OF THE MULTI-GRAM STRATEGY

To obtain better relation indication information, we conduct experiments to choose an effective multi-gram strategy.

We first only choose tri-gram, four-gram, and five-gram and we obtain the corresponding classification F1 value as 83.5%, 83.2%, and 84.0%, respectively. From Table 2, we can see that, for frameworks that only utilize one *n*-gram-based strategy, a five-gram-based strategy can obtain better relation indication information than tri- and four-gram-based strategies in this corpus. It is also obvious that the multi-gram strategies can improve relation classification performances from one *n*-gram strategies. This is mainly because the CNN model with multi-grams can allow patterns of different lengths to be learned. The multi-gram strategy of tri-gram, four-gram, and five-gram (3+4+5) can improve the F1 value to 84.7% from the multi-gram strategy of tri-gram and four-gram (3+4) with an F1 value of 84.3%. The F1 value of the multi-gram strategy of tri-gram, four-gram, five-gram, and six-gram (3+4+5+6) shows that we cannot improve the final classification performance by using too many *n*-grams. All the above experiment results illustrate that the adoption of a multi-gram strategy of tri-gram, four-gram, and five-gram can extract more effective relation indication information than other multi-gram strategies.

TABLE 2. Multi-gram strategy settings.

CNN Window length	F1 value (%)
3	83.5
4	83.2
5	84.0
3+4	84.3
3+4+5	84.7
3+4+5+6	84.6

C. IMPACT OF THE DROPOUT STRATEGY

We now validate the dropout strategies proposed in Section III-E. We first drop out the word embeddings layer, then with a fixed dropout rate of the word embeddings layer, we test the effects of dropping out the attention layer and the penultimate units, respectively.

We find that the dropout of the embeddings layer improves model performance by 1.04% (Figure 7a); dropout of the penultimate layer further improves performance by 3.19% (Figure 7b); dropout of the attention layer hurts the final F1 value after the dropout rate becomes larger than 0.2, and the best result can improve model performance by 0.87% with the dropout rate of 0.2 (Figure 7c). This analysis can also provide some clues for dropout in other studies of attention models in GRU frameworks.

[1] Available at: http://code.google.com/p/word2vec/

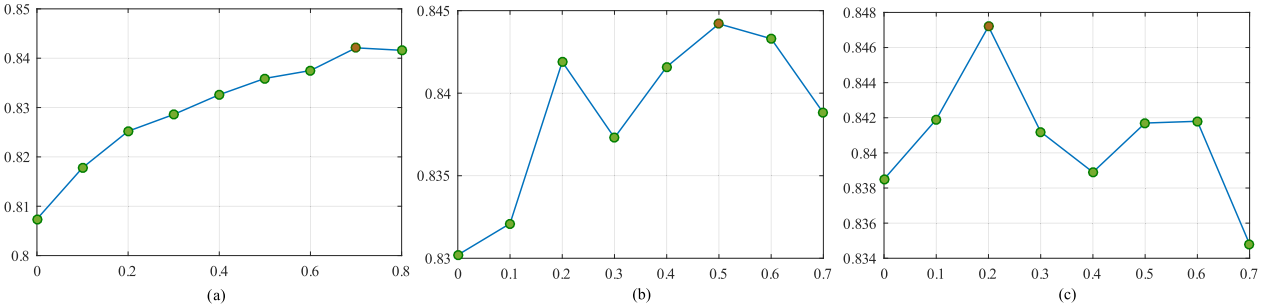

FIGURE 7. F1 scores versus dropout rates. We first evaluate the effect of dropout embeddings (a). Then the dropout of the penultimate layer (b) and the attention layer (c) is tested with word embeddings being dropped out by 0.7.

D. IMPACT OF THE MULTI-GRAM CNN-BASED SELF-ATTENTION MODEL

To show the effectiveness of the multi-gram CNN-based self-attention model, we give a detailed comparison of the results of variants of our proposed model in Table 3. Here, we label operations of max-pooling and sum to produce sentence representation as MaxPooling and Sumand we label the most frequently used random attention model proposed in [33] as Random-Att-BiLSTM. Our proposed multi-gram CNN-based self-attention model is labeled as MCNN-Att-BiGRU.

TABLE 3. Comparison between the main model and variants.

Model	F1(%)
BiGRU+MaxPooling	83.5
BiGRU+Sum	83.4
BiGRU+Random-Att+MaxPooling	84.0
BiGRU+Random-Att+Sum	84.2
BiGRU+MCNN-Att+MaxPooling	84.5
BiGRU+MCNN-Att+Sum	**84.7**

From Table 3, we can see that F1 values of attention-based models are all above 84.0%, which indeed improve the classification performance from models not using the attention method. In particular, compared with the random attention strategy, our proposed multi-gram CNN-based self-attention model gives a better performance with an F1 value of 84.7%. From this point of view, we can conclude that our proposed attention strategy is more effective than the traditional random attention model for the relation classification task.

Figure 8 illustrates the attention value visualization of four relation types by using the multi-gram CNN-based self-attention model. For instance, histogram (a) shows the weight distribution over an annotated sentence of relation *Instrument–Agency (e2,e1)*. According to the human understanding, the words "proposed," "assessing," and "complements" play important roles in identifying the targeted relation type. As expected, our multi-gram CNN-based self-attention model assigns conspicuous weight for "going," "away," and "from" for the example sentence of relation *Entity–Origin (e2,e1)* in histogram (b). It is more obvious for the example sentence of relation *Message–Topic (e1,e2)* in histogram (d). Its sentence length is 21, whereas the relation-related features are only eight words including the target entity pair. Our proposed attention model assigns larger weight for words with crucial information such as

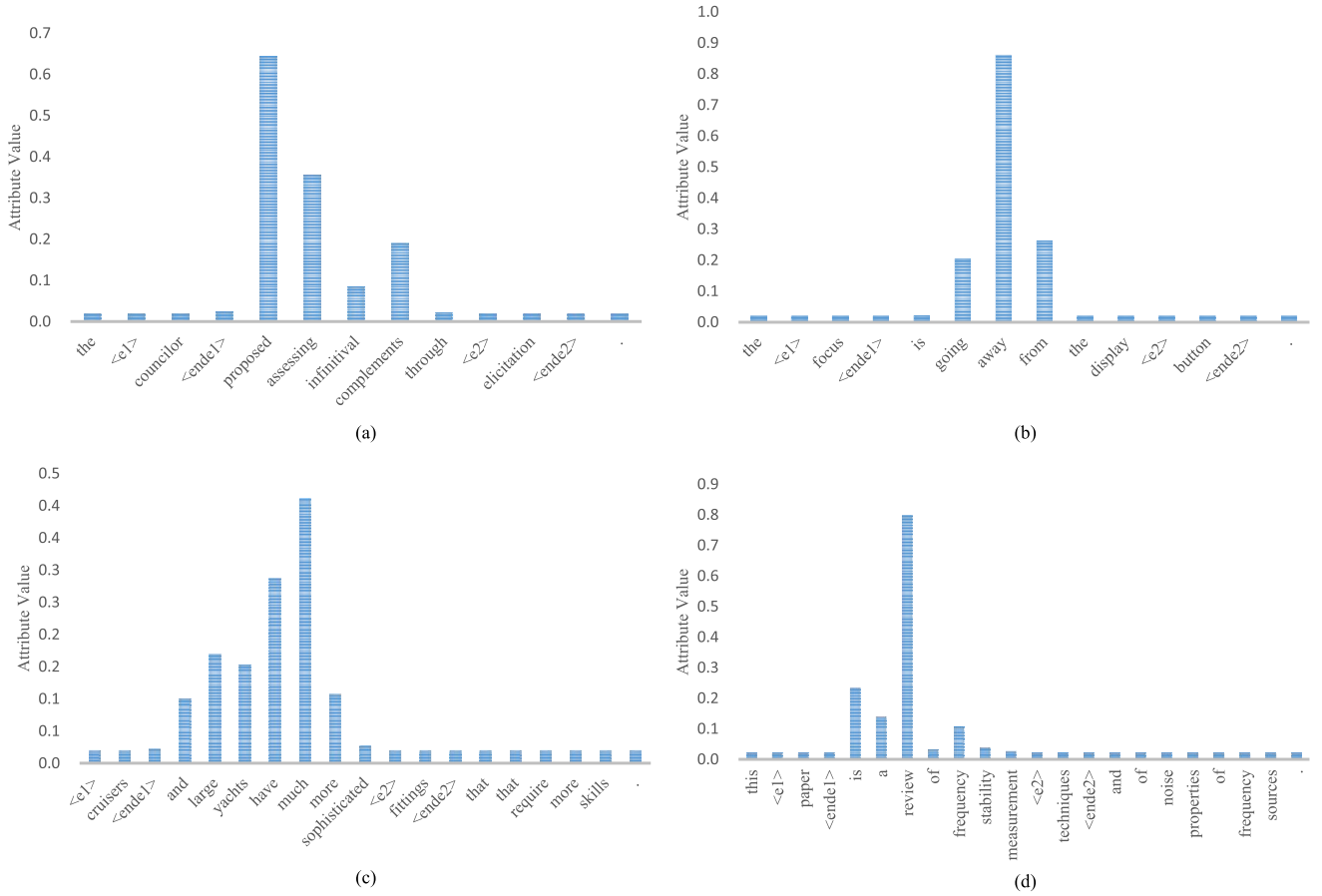

FIGURE 8. Attention visualization of the multi-gram CNN-based self-attention model in four relation types: *Instrument–Agency*, *Entity–Origin*, *Component–Whole*, and *Message–Topic*. To give a better visual feeling, the value of the y-coordinate is computed as $\exp(a_i) - 0.98$. (a) Instrument-agency (e2,e1). (b) Entity-origin (e2,e1). (c) Component-whole (e2,e1). (d) Message-topic (e1,e2).

TABLE 4. Comparison with previous relation classification systems. PF denotes position feature, PI stands for position indicator.

Model Information	Additional	F1(%)
SVM [9]	POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner	82.2
MVRNN [17]	word embedding, syntactic parsing tree +POS, NER, WordNet	79.1 82.4
CNN [10]	Word embeddings+PF +WordNet, words around nominal	78.9 82.7
BRNN [76]	Word embeddings	82.5
CR-CNN [22]	Word embeddings + PF	82.8 84.1
SDP-LSTM [25]	Word embeddings +POS+GR+WordNet embeddings	82.4 83.7
BLSTM [19]	Word embeddings +PF+POS+NER+WNSYN+DEP	82.7 84.3
Random-Att-BLSTM [33]	Word embeddings+PI	84.0
MCNN-Att-BiGRU	Word embeddings+PI+SDP	**84.7**

"proposed" for *Instrument–Agency (e2,e1)*, "away" for *Entity–Origin (e2,e1)*, "much" for *Entity–Origin (e2,e1)*, and "review" for *Message–Topic (e1,e2)*. All facts mentioned above indicate that the proposed attention model can locate relation trigger features and only assign importance weights for them. This means that the assigned attention weights are very concentrated. Because words not related to target relations are not assigned importance weights whereas importance weights of relation-related features have significant distinction between each other. This is a good characteristic of our attention model.

E. COMPARISON WITH PREVIOUS RELATION CLASSIFICATION MODELS

Table 4 compares our proposed model with other state-of-the-art methods of relation classification. From Table 4, we can see that, except for the first method (SVM method), all other methods are neural-network-based relation classification methods. Our proposed method is based on BiGRU network, which also belongs to deep neural network frameworks.

For the SemEval-2010 Task 8 benchmark, the SVM [9] presented in the first entry is the top performing traditional feature-based method. This model combines with a rich set of costly handcrafted features to generate sentence-level features, and achieves an F1 score of 82.2%.

Subsequently, more progress has been made by deep neural network frameworks. MVRNN [17] pioneered the use of end-to-end neural networks for relation classification. It constructs a recursive neural network based on the syntactic parsing tree and simultaneously trains additional matrices to modify the meanings of neighboring words. It raises the F1 score to 82.4%. CNN [10] leverages the raw word sequence as input and exploits position features to identify the position of the entity pair. The extra lexical features are transformed into distributed representations and concatenated with sentence-level vectors to predict relation classes. Hence, the F1 score is increased to 82.7%. CR-CNN [22] focuses more on the influence of the class Other, which proposes a new pairwise ranking function to substitute softmax. This targeted modification obtains an F1 score of 84.1%. However, our proposed model can achieve a superior F1 score of 84.7%.

Among all these state-of-the-art methods, BRNN, SDP-LSTM, BLSTM, and Random-Att-BLSTM model are the four most relevant works to our model.

Similar to our proposed model, BRNN and BLSTM utilize a bidirectional RNN architecture. BRNN leverages the original RNN and max-pooling operation to extract sentence-level features, and achieves an F1 score of 82.5% without a position indicator. BLSTM adopts bidirectional LSTM to model sequence learning, and employs a piecewise max-pooling to generate a sentence-level representation. It achieves an F1 score of 82.7%. SDP-LSTM [25] treats the shortest dependency path between the entity pair as input to pick up heterogeneous information. The external linguistic features are integrated via multi-channel LSTM networks. Based on this, it achieves an F1 score of 83.7%. The improvement shows that the utilization of the shortest dependency path can enhance the final classification performance. Inspired by this clue, our proposed model attempts to model sentence-level features on a BiGRU framework by using a multi-gram CNN-based self-attention mechanism. The self-attention mechanism utilizes a multi-gram CNN model to extract relation indicate information from the shortest dependency path between the two mentioned entities. Our proposed model can achieve an F1 score of 84.7%, much higher than the BRNN, BLSTM, and SDP-LSTM models. Based on the conclusion obtained in Section IV-D, the good performance is mainly generated by the utilization of the self-attention mechanism.

To further compare our proposed attention-based model with existing attention-based models, we compare our model with the Random-Att-BLSTM. Random-Att-BLSTM [33] adopts a random attention model, which randomly initializes the attention vector, and learns it by a general end-to-end deep neural network framework without using any relation indication information. It achieves an F1 score of 84%. The disadvantage of Random-Att-BLSTM is that it cannot make use of

TABLE 5. Structure of the small dataset.

Relation Number	Relation Name	No. of training sentence	No. of testing sentence
1	/location/neighborhood/neighborhood_of	7630	1880
2	/business/company/founders	842	238
3	/people/person/place_of_birth	3466	820
4	/people/deceased_person/place_of_death	2047	500
5	/location/us_state/capital	667	154
6	/location/administrative_division/country	7377	1891
7	/people/person/nationality	9812	2296
8	/business/person/company	5608	1450
9	/people/person/place_lived	7793	1993
10	/location/country/capital	9403	2312
11	/location/location/contains	8340	2160
12	/location/country/administrative_divisions	7367	1900

relation indication information included in the inputs. Rather than randomly initialize the attention vector, our proposed model designs a multi-gram CNN-based self-attention model in a BiGRU framework to extract relation indication information from the shortest dependency path of the two entities, and the extracted relation indication information is used to supervise the generation of the attention vector. This mean that our proposed model can adaptively make use of relation indication information in a general framework. It can overcome the disadvantage of the Random-Att-BLSTM model, and obtain better performance than the Random-Att-BLSTM model with an F1 score of 84.7%.

F. EXPERIMENTS ON A SMALL DATASET

The second dataset is a small dataset which is taken from a widely used dataset (New York Times (NYT) corpus) in distant supervised methods [36]–[38]. The NYT corpus was developed by Riedel *et al.* [76] by aligning Freebase5 relations with the NYT corpus of the years 2005–2007. There are 53 possible relationships including a special relation NA, which indicates there is no relation between head and tail entities.

As it is obtained by heuristically aligning an already existing knowledge base to texts, the heuristic alignment can fail, resulting in the wrong label problem. In addition, the data size of the 53 relations in the dataset is quite different. This means that the NYT corpus dataset is a rough dataset with lots of noisy and serious category imbalance problem. In this paper, to simplify the classification problem on the noisy dataset, we classify relations by regarding relation bags as units (entity pair equals relation bag). We choose 12 relations whose data sizes are equivalent to form a small dataset. The structure of the small dataset is illustrated in Table 5.

To investigate the performance of our proposed model on a noisy dataset, we apply it on the small dataset and compare it with some related state-of-the-art methods. Parameter settings are similar to Section IV-A. Experimental results are listed in Tables 6 and 7.

Table 6 illustrates the detail classification results of each relation. From Table 6, we can see that for most relations, our proposed model can obtain good classification results. It is also obvious that classification results

TABLE 6. Detailed classification results of each relation.

Relation Number	Precision(%)	Recall(%)	F1(%)
1	99	100	99.5
2	99	94	96.4
3	51	36	42.2
4	43	51	46.7
5	46	79	58.1
6	99	99	99.0
7	93	93	93.0
8	95	97	95.9
9	79	83	80.9
10	60	97	74.1
11	95	58	72.0
12	83	50	62.4

TABLE 7. Comparisons with three related traditional models. The precision, recall, and F1 values are the overall results of all 12 relations.

Model	Precision(%)	Recall(%)	F1(%)
BiGRU	77	76	76.4
SDP-GRU	82	81	81.5
Random-Att-BiGRU	83	81	81.9
MCNN-SDP-Att-BiGRU	84	82	82.9

of different relations are quite different. Some relations, such as */location/neighborhood/neighborhood_of*, */location/administrative_division/country*, and */business/person/company*, can achieve an F1 value of at least 95%, whereas for some relations such as */people/person/place_of_birth* and */people/deceased_person/place_of_death*, their F1 values are only 42.2% and 46.7%. The main reason that leads to the different results is the quality of the training dataset. As we classify relations by regarding bags as units, the datasize of bags is very important. Take */people/deceased_person/place_of_death* as an example, there are 1540 entity pairs with a corresponding 3466 sentences in its training data. However, there is one entity pair $< saddam\_hussein, iraq >$, whose bag includes 680 sentences and most of them are just entity-pair mentions but not relation mentions, whereas each of the remaining entity pairs only contains no more than 50 sentences. The unbalance distribution and poor quality of training data lead to unsatisfactory results.

From Table 7, we can see that the BiGRU model has the poorest performance. Unlike the most frequently used SemEval-2010 Task 8 benchmark in supervised methods,

the small dataset includes much more noise, so it may introduce more noisy information, and lead to poor performance by only using BiGRU framework. The SDP-GRU model can improve the classification results largely from the BiGRU model with an F1 value of 81.5%. The improvement is obtained by using the shortest dependency path features, because it can not only remove noise from inputs, but can also condense the most illuminating information for entities' relation. The Random-Att-BiGRU model can obtain better results than the BiGRU model and SDP-GRU model. This means that the utilization of the attention mechanism is effective. Our proposed model can obtain the best classification result with an F1 value of 82.9%, which is better than the result of the Random-Att-BiGRU model. The main reason for the improvement is that our proposed model utilizes the multi-gram CNN-based self-attention framework to learn relation indicate information from the shortest dependency path of inputs. Hence, compared with the Random-Att-BiGRU model, it can better locate relation trigger features and further improve the final classification performance.

V. CONCLUSION

We have introduced a novel multi-gram CNN-based self-attention model on a general RNN network in this paper. The multi-gram CNN-based self-attention framework can extract relation indication information from the shortest dependency path of the two mentioned entities. Based on the extracted relation indication information, our proposed model can generate the attention vector and obtain the ''importance'' distribution over inputs by computing the similarity of each word with the relation attention vector. Experimental results show that our proposed model is able to locate crucial information of relations and can overcome the disadvantages of existing attention models. Most importantly, the proposed model also has a good characteristic that it can obtain more concentrated attention weights compared with other attention models. We expect this sort of architecture to be used in other tasks, which can be explored in future work.

REFERENCES

[1] F. Wu and D. S. Weld, ''Open information extraction using Wikipedia,'' in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 118–127.

[2] X. Yao and B. Van Durme, ''Information extraction over structured data: Question answering with freebase,'' in *Proc. ACL*, vol. 1, 2014, pp. 956–966.

[3] Y. Xu, G. Li, L. Mou, and Y. Lu, ''Learning non-taxonomic relations on demand for ontology extension,'' *Int. J. Softw. Eng. Knowl. Eng.*, vol. 24, no. 8, pp. 1159–1175, 2014.

[4] R. Grishman and B. Sundheim, ''Message understanding conference-6: A brief history,'' in *Proc. COLING*, vol. 96, 1996, pp. 466–471.

[5] (2008). *Automatic Content Extraction*. [Online]. Available: http://www.nist.gov/speech/tests/ace

[6] KBP. (2013). *Knowledge Base Population 2013*. [Online]. Available: http://www.nist.gov/tac/2013/KBP/

[7] (2013). *TAC KBP 2013: English Slot Filling—Regular and Temporal*. [Online]. Available: http://surdeanu.info/kbp2013/def.php

[8] R. C. Bunescu and R. J. Mooney, ''A shortest path dependency kernel for relation extraction,'' in *Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 724–731.

[9] B. Rink and S. Harabagiu, ''UTD: Classifying semantic relations by combining lexical and semantic resources,'' in *Proc. 5th Int. Workshop Semantic Eval.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 256–259.

[10] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, ''Relation classification via convolutional deep neural network,'' in *Proc. COLING*, 2014, pp. 2335–2344.

[11] N. Kambhatla, ''Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations,'' in *Proc. ACL Interact. Poster Demonstration Sessions*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, Art. no. 22.

[12] F. M. Suchanek, G. Ifrim, and G. Weikum, ''Combining linguistic and statistical analysis to extract relations from Web documents,'' in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 712–717.

[13] T. H. Nguyen and R. Grishman, ''Employing word representations and regularization for domain adaptation of relation extraction,'' in *Proc. ACL*, vol. 2, 2014, pp. 68–74.

[14] L. Qian, G. Zhou, F. Kong, Q. Zhu, and P. Qian, ''Exploiting constituent dependencies for tree kernel-based semantic relation extraction,'' in *Proc. 22nd Int. Conf. Comput. Linguistics*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 697–704.

[15] R. C. Bunescu and R. J. Mooney, ''Subsequence kernels for relation extraction,'' in *Proc. NIPS*, 2005, pp. 171–178.

[16] 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.

[17] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, ''Semantic compositionality through recursive matrix-vector spaces,'' in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 1201–1211.

[18] Y. Xu *et al.* (2016). ''Improved relation classification by deep recurrent neural networks with data augmentation.'' [Online]. Available: https://arxiv.org/abs/1601.03651

[19] S. Zhang, D. Zheng, X. Hu, and M. Yang, ''Bidirectional long short-term memory networks for relation classification,'' in *Proc. PACLIC*, 2015, pp. 73–78.

[20] C. Cui, H. Liu, T. Lian, L. Nie, L. Zhu, and Y. Yin, ''Distribution-oriented aesthetics assessment with semantic-aware hybrid network,'' *IEEE Trans. Multimedia*, to be published.

[21] T. H. Nguyen and R. Grishman, ''Relation extraction: Perspective from convolutional neural networks,'' in *Proc. NAACL-HLT*, 2015, pp. 39–48.

[22] C. N. dos Santos, B. Xiang, and B. Zhou, ''Classifying relations by ranking with convolutional neural networks,'' *Comput. Sci.*, vol. 86, no. 86, pp. 132–137, 2015.

[23] K. Xu, Y. Feng, S. Huang, and D. Zhao, ''Semantic relation classification via convolutional neural networks with simple negative sampling,'' *Comput. Sci.*, vol. 71, no. 7, pp. 9–941, 2015.

[24] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, and H. Wang, ''A dependency-based neural network for relation classification,'' *Comput. Sci.*, Jun. 2015.

[25] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, ''Classifying relations via long short term memory networks along shortest dependency paths,'' in *Proc. EMNLP*, 2015, pp. 1785–1794.

[26] D. Tang, B. Qin, and T. Liu, ''Document modeling with gated recurrent neural network for sentiment classification,'' in *Proc. EMNLP*, 2015, pp. 1422–1432.

[27] M. Miwa and M. Bansal, ''End-to-end relation extraction using LSTMs on sequences and tree structures,'' in *Proc. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1105–1116.

[28] K. Hashimoto, M. Miwa, Y. Tsuruoka, and T. Chikayama, ''Simple customization of recursive neural networks for semantic relation classification,'' in *Proc. EMNLP*, 2013, pp. 1372–1376.

[29] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' *Comput. Sci.*, Sep. 2014.

[30] A. M. Rush, S. Chopra, and J. Weston, ''A neural attention model for abstractive sentence summarization,'' *Comput. Sci.*, Sep. 2015.

[31] C. dos Santos, M. Tan, B. Xiang, and B. Zhou. (2016). ''Attentive pooling networks.'' [Online]. Available: https://arxiv.org/abs/1602.03609

[32] L. Wang, Z. Cao, G. de Melo, and Z. Liu, ''Relation classification via multi-level attention CNNs,'' in *Proc. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1298–1307.

[33] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, p. 207.
[34] L. Cong and X. Minguang, "Semantic relation classification via hierarchical recurrent neural network with attention," in *Proc. COLING 26th Int. Conf. Comput. Linguistics*, 2016, pp. 1254–1263.
[35] P. Qin, W. Xu, and J. Guo, "Designing an adaptive attention mechanism for relation classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 4356–4362.
[36] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proc. ACL*, vol. 1, 2016, pp. 2124–2133.
[37] G. Ji, K. Liu, S. He, and J. Zhao, "Distant supervision for relation extraction with sentence-level attention and entity descriptions," in *Proc. AAAI*, 2017, pp. 3060–3066.
[38] L. Yang, T. L. J. Ng, C. Mooney, and R. Dong, "Multi-level attention-based neural networks for distant supervised relation extraction," in *Proc. 25th Irish Conf. Artif. Intell. Cogn. Sci.*, 2017, pp. 1–12.
[39] Y. Lu, C. Zhang, and W. Xu, "Instance-adaptive attention mechanism for relation classification," in *Proc. Int. Conf. Artif. Neural Netw.*, 2017, pp. 322–330.
[40] Z. GuoDong, S. Jian, Z. Jie, and Z. Min, "Exploring various knowledge in relation extraction," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 427–434.
[41] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *J. Mach. Learn. Res.*, vol. 3, pp. 1083–1106, Feb. 2003.
[42] M. Wang, "A re-examination of dependency path kernels for relation extraction," in *Proc. IJCNLP*, 2008, pp. 841–846.
[43] B. Plank and A. Moschitti, "Embedding semantic similarity in tree kernels for domain adaptation of relation extraction," in *Proc. ACL*, vol. 1, 2013, pp. 1498–1507.
[44] N. Bach and S. Badaskar, "A review of relation extraction," in *Literature review for Language and Statistics II*. 2007.
[45] N. T. Vu, H. Adel, P. Gupta, and H. Schütze, "Combining recurrent and convolutional neural networks for relation classification," in *Proc. NAACL*, 2016.
[46] Z. Ma, J.-H. Xue, A. Leijon, Z.-H. Tan, Z. Yang, and J. Guo, "Decorrelation of neutral vector variables: Theory and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 129–143, Jan. 2018.
[47] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[48] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[49] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, Aug. 2002.
[50] Z. Ma, H. Yu, W. Chen, and J. Guo, "Short utterance based speech language identification in intelligent vehicles with time-scale modifications and deep bottleneck features," *IEEE Trans. Veh. Technol.*, to be published.
[51] P. F. Sepp Hochreiter, Y. Bengio, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Networks*, S. C. Kremer and J. F. Kolen, Eds. Piscataway, NJ, USA IEEE Press, 2001.
[52] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.
[53] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
[54] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2625–2634.
[55] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with LSTM recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3547–3555.
[56] Z. Ma, Y. Lai, W. B. Kleijn, Y.-Z. Song, L. Wang, and J. Guo, "Variational Bayesian learning for Dirichlet process mixture of inverted Dirichlet distributions in non-Gaussian image feature modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.
[57] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck. (2016). "Contextual LSTM (CLSTM) models for large scale NLP tasks." [Online]. Available: https://arxiv.org/abs/1602.06291
[58] M.-T. Luong and C. D. Manning. (2016). "Achieving open vocabulary neural machine translation with hybrid word-character models." [Online]. Available: https://arxiv.org/abs/1604.00788
[59] M. Yang, W. Tu, J. Wang, F. Xu, and X. Chen, "Attention based LSTM for target dependent sentiment classification," in *Proc. AAAI*, 2017, pp. 5013–5014.
[60] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, "Variational Bayesian matrix factorization for bounded support data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 876–889, Apr. 2015.
[61] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Comput. Sci.*, Sep. 2014.
[62] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.
[63] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[64] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Comput. Sci.*, Sep. 2015.
[65] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom. (2015). "Reasoning about entailment with neural attention." [Online]. Available: https://arxiv.org/abs/1509.06664
[66] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *Comput. Sci.*, Dec. 2014.
[67] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. NAACL-HLT*, 2016, pp. 1480–1489.
[68] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Comput. Sci.*, Sep. 2013.
[69] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Auton. Secure Comput., Pervasive Intell. Computing (CIT/IUCC/DASC/PICOM)*, Oct. 2015, pp. 2359–2364.
[70] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Trans. Assoc. Comput. Linguistics*, vol. 2, no. 1, pp. 207–218, 2013. [Online]. Available: https://Nlp.stanford.edu
[71] G. Hinton, N. Srivastava, A. Krizhevsky, R. R. Salakhutdinov, and I. Sutskever, "Improving neural networks by preventing co-adaptation of feature detectors," *Comput. Sci.*, vol. 3, no. 4, pp. 212–223, Jul. 2012.
[72] T. Cai and W.-X. Zhou, "A max-norm constrained minimization approach to 1-bit matrix completion," *J. Mach. Learn. Res.*, vol. 14, pp. 3619–3647, Dec. 2013.
[73] I. Hendrickx *et al.*, "SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proc. Workshop Semantic Eval., Recent Achievements Future Directions*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 94–99.
[74] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *Comput. Sci.*, Dec. 2012.
[75] D. Zhang and D. Wang, "Relation classification via recurrent neural network," *Comput. Sci.*, Dec. 2015.
[76] S. Riedel, L. Yao, and A. Mccallum, "Modeling relations and their mentions without labeled text," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2010, pp. 148–163.

CHUNYUN ZHANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2015. She is currently a Lecturer with the School of Computer Science and Technology, Shandong University of Finance and Economics. Her main research interests include natural language processing, information extraction, and data mining.

CHAORAN CUI received the B.S. degree in software engineering and the Ph.D. degree from Shandong University, Jinan, China, in 2010 and 2015, respectively. He was a Research Fellow with Singapore Management University, from 2015 to 2016. He is currently a Professor with the School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan. His research interests include information retrieval, analysis and understanding of multimedia information, and computer vision.

SHENG GAO received the Ph.D. degree from the Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie, in 2012. He is currently an Assistant Professor with the Beijing University of Posts and Telecommunications. His research interests include machine learning and data mining, social network analysis, knowledge mapping, information recommendation, and biological information processing.

XIUSHAN NIE received the Ph.D. degree from Shandong University, Jinan, China, in 2011. From 2013 to 2014, he was a Visiting Scholar with the University of Missouri, Columbia, MO, USA. He is currently a Professor with the Shandong University of Finance and Economics, Jinan. His research interests include data mining, multimedia retrieval, indexing, and computer vision.

WEIRAN XU received the Ph.D. degree from the Beijing University of Posts and Telecommunications, China, in 2003, where he is currently an Associate Professor with the Web Searching Teaching and Research Center. His current research interests include information extraction, pattern recognition, and machine learning.

LU YANG received the Ph.D. degree from Shandong University, in 2016. She is currently a Lecturer with the School of Computer Science and Technology, Shandong University of Finance and Economics. Her main research interests include finger vein recognition and biometrics.

XIAOMING XI received the Ph.D. degree from Shandong University, in 2015. He is currently a Lecturer with the School of Computer Science and Technology, Shandong University of Finance and Economics. His main research interests include biometrics, medical image processing, and machine learning.

YILONG YIN received the Ph.D. degree from Jilin University, Changchun, China, in 2000. From 2000 to 2002, he was a Postdoctoral Fellow with the Department of Electronic Science and Engineering, Nanjing University, Nanjing, China. He is currently the Director of the Machine Learning and Data Mining Laboratory and a Professor with Shandong University, Jinan, China. His research interests include machine learning, data mining, and biometrics.